08-21-00

A

FISH & RICHARDSON P.C.

225 Franklin Street
Boston, Massachusetts
02110-2804

Telephone
617 542-5070

Facsimile
617 542-8906

Web Site
www.fr.com

jc841 U.S. PTO
09/642268
08/18/00

Frederick P. Fish
1855-1930

W.K. Richardson
1859-1951

JC893 U.S. PTO
08/18/00

BOSTON
DALLAS
DELAWARE
NEW YORK
SAN DIEGO
SILICON VALLEY
TWIN CITIES
WASHINGTON, DC

August 18, 2000

Attorney Docket No.: 07072-935001

Box Patent Application
Commissioner for Patents
Washington, DC 20231

Presented for filing is a new original patent application of:

Applicant: MAUREEN A. LALLY, KENNETH R. GOGUEN, PETER C. LAUTERBACH

Title: GRAPHICAL USER INPUT INTERFACE FOR TESTING PERFORMANCE OF A MASS STORAGE SYSTEM

Enclosed are the following papers, including those required to receive a filing date under 37 CFR 1.53(b):

	Pages
Specification	14
Claims	1
Abstract	1
Declaration	2
Drawing(s)	28

Enclosures:
- — Assignment cover sheet and an assignment, 3 pages, and a separate $40 fee.
- — Postcard.

Basic filing fee	$690
Total claims in excess of 20 times $18	$0

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL 624 320 884 US

I hereby certify under 37 CFR §1.10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

August 18, 2000
Date of Deposit

Samantha Bell
Signature

Samantha Bell
Typed or Printed Name of Person Signing Certificate

Independent claims in excess of 3 times $78	$0
Fee for multiple dependent claims	$0
Total filing fee:	$690

A check for the filing fee is enclosed. Please apply any other required fees or any credits to deposit account 06-1050, referencing the attorney docket number shown above.

If this application is found to be incomplete, or if a telephone conference would otherwise be helpful, please call the undersigned at (617) 542-5070.

Kindly acknowledge receipt of this application by returning the enclosed postcard.

Please send all correspondence to:

GARY A. WALPERT
Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110-2804

Respectfully submitted,

Gary A. Walpert
Reg. No. 26,098 Reg No. 30,175
Enclosures
GAW/rll
20118391.doc

Attorney's Docket No.: 07072-935001 / EMC 99-172

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: GRAPHICAL USER INPUT INTERFACE FOR TESTING PERFORMANCE OF A MASS STORAGE SYSTEM

APPLICANT: MAUREEN A. LALLY, KENNETH R. GOGUEN AND PETER C. LAUTERBACH

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. type label no. EL 624 320 884 US

I hereby certify under 37 CFR §1.10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

August 18, 2000
Date of Deposit

Samantha Bell
Signature

Samantha Bell
Typed or Printed Name of Person Signing Certificate

GRAPHICAL USER INPUT INTERFACE FOR TESTING PERFORMANCE OF A MASS STORAGE SYSTEM

BACKGROUND OF THE INVENTION

The invention relates generally to tools and systems for measuring the performance of mass storage systems, and more particularly, to methods and apparatus for developing, measuring, analyzing, and displaying the performance statistics of a plurality of disk drive elements controlled through a disk drive controller connected to a plurality of host computers.

As the size and complexity of computer systems increase, including the number of host computers and the number of disk drive elements, it becomes increasingly important to measure and understand the functions and parameters which affect the performance of the system. The performance of the system can be typically measured in terms of input/output (I/O) response times, that is, the time it takes for a read or write command to be acted upon, as far as the host computer is concerned, by the disk drive controller system.

It is well known, in the field, to measure, usually using a single parameter, the instantaneous or average response time of the system. Typically, a host computer outputs one or more I/O requests to the disk drive controller, and then measures the time for a response to be received from the disk drive controller. This time duration, while representative of the response of a specific read or write command to the disk drive system, is most often not representative of the actual performance which can be obtained from the system.

A similar distortion, not representative of system performance, can occur when average response time values are determined. For example, a disk controller, using a cache memory in its write process, can have substantially different write time responses depending upon the availability of cache memory. An average response (the average of, for example, a write where cache was available and one where cache was not available) would be misleading and meaningless.

The performance of a large storage system is particularly difficult to measure since more than one of the host computers, which connect to be disk drive controller(s), can

operate at the same time, in a serial or in a parallel fashion. As a result, a plurality of disk drive elements, usually arranged in a disk drive array, operating in either an independent fashion, a RAID configuration, or a mirrored configuration, for example, can have a significant yet undetectable bandwidth or operational problem which cannot be addressed, or discovered, when commands are sent only from a single host computer.

In U.S. Patent 5,953,689, issued September 14, 1999, assigned to the assignee of this application, an improved method of time synchronizing a plurality of hosts operating in a variety of different configurations, and of issuing commands according to a prescribed sequence, was described. The method described in the above-identified patent features sending test requests, for the mass storage system, from a "main" host computer to each of a plurality of client host computers, executing at each host computer a test request sequence by sending commands to the mass storage system, accumulating at each host computer data regarding performance of the mass storage system, the data being in response to the requests or commands sent by each particular host computer, and sending, from each host computer to a master host computer, data regarding the performance of the mass storage system in response to the host generated commands. Significant data reduction techniques control and organize the data for later analysis.

While this system worked well, it had, at the time the application for the system was filed, specific limitations with regard to creating the test configurations and with regard to the flexibility of the analysis and processing once the statistics had been collected.

SUMMARY OF THE INVENTION

The invention relates to an improved method for measuring the system performance of a mass storage system having a plurality of disk drive storage elements controlled by a disk drive controller. Typically the disk drive controller has a cache memory. The controller receives commands and data from, and returns at least data to, typically, a plurality of host computers.

A method for measuring system performance in a mass storage system, the storage system having a plurality of disk drive storage elements controlled by a disk drive controller, said controller receiving commands and data from and returning at least data to a plurality of host computers, features enabling a graphical user interface for generating an input parameter

containing sequence input of commands for operating the system for measuring system performance, generating from the input parameter sequence a test sequence input identifying commands to be sent to the storage system, executing at at least one host computer a test request identified by the test sequence input, by sending commands to the mass storage system, accumulating, at at least the executing host computer, data regarding performance of the mass storage system, in response to the requests sent by the host computer, and processing the accumulated data regarding the performance of the mass storage system in response to the one host-generated commands. In one aspect, the processing features validating and correcting, as required, the accumulated data.

The invention further features generating, at the graphical user interface, configuration files, workbench files, and/or benchmark files. In another aspect, the invention features selecting, using the graphical user interface, from various test types for the input sequence of commands, in a point and click fashion. The test types can include defining a system configuration, test periods, and a sequence of test repeat parameters.

Advantageously, therefore, the system easily, and in large part automatically, develops, measures, generates and analyzes statistics describing the dynamic performance of the mass storage system, from the host computers to the disk drive elements, wherein individual operations as well as sequences of operations can easily be set up, with selected initial conditions, and accurately and automatically tested according to desired criteria using graphical user interfaces. The method and apparatus of the invention further advantageously enable the user to easily configure, set, and determine, using a graphical user interface, read/write sequencing and a relative mix of read and write commands, as well as enabling the dynamic performance of the system to be repeatedly enabled and tested for consistency, accuracy, and effectiveness.

DESCRIPTION OF DRAWINGS

Other features and advantages of the invention will be apparent from the following description, taken together with the drawings, in which:

Figure 1 shows a typical system in which the invention is useful;

Figure 2 shows, in more detail, a particular controller system in which the invention finds particular use;

Figure 3 is a flow chart showing overall operation of the setup/analysis portion of the system;

Figure 4 shows a table of arguments used in the system testing;

Figure 5 shows a more detailed flow chart in accordance with the operation of the system;

Figure 6 shows the operational subsystems of the system;

Figure 7 shows data reduction sequential flow;

Figures 8A, 8B, 8C, and 8D show examples of trends data presentation;

Figure 9 shows screen shots of the graphical user interface of the system for Post Test Data Processing;

Figures 9A-9O show screen shots of the graphical user interface of the system;

Figure 10 shows a flow chart illustrating post processing;

Figure 10A-10B show screen shots of the post processing presentation;

Figure 11 illustrates various system database files; and

Figure 12 shows a screen shot of the trends analysis presentation.

DETAILED DESCRIPTION

Referring to Figure 1, the invention relates to a computer system wherein a plurality of host computers or processors 12a, 12b, ..., 12n, connect to a storage controller system 14, such as the EMC Symmetrix® storage system. The controller acts as an intelligent interface between the host computers and a plurality of mass storage devices, such as, for example, disk drive elements 16a, 16b, ..., 16k. Data written by the host or read from the disk drive elements passes through the memory controller system which acts as a two way communications path with substantial capabilities. The disk drive elements can have any of, or a combination of, a plurality of configurations. For example, in some systems, the data from a host is uniformly striped across all of the disk storage devices; and in other systems, the data from a host is stored on the disk drives 16 according to a RAID protocol or an n-way mirrored protocol. In yet other embodiments of the invention, all of the data from a particular host may be stored in a logical volume on a single disk drive or allocated to different logical volumes of the same or different disk drives, depending upon the nature and

the source of the data and host. A host computer can also read data from one or more of the disk drive units to generate a single host logical volume.

To determine the limits of performance in the system, the hosts can, according to the invention, be operated to exercise and test the memory controller and the disk drive elements. Thus potential problems which can create a bottleneck on those communication lines connected from the disk drive controller to either the disk drive elements or the hosts can be identified, as can cache memory loading issues in the drive controller.

Referring to Figure 2, in a particular embodiment according to the invention, the disk controller has a plurality of host adaptors (also referred to as channel directors, FA's or SA's) 30 connecting to a global memory 32 through which, in this embodiment, all data and commands flow. The global memory 32 is connected to a plurality of disk adaptors (also referred to as DA's or disk directors) 34 which connect the disk drives 16 to storage 32 through drive ports 35 of the adaptors 34 over lines 39. In accordance with this particular embodiment of the invention, each host adaptor has a SCSI adaptor embedded therein which communicates with the global memory 32. In the illustrated embodiment, the read and write operations pass through each SCSI adaptor unit 34 and to the disk adaptors to the disk drive elements. Each host adaptor connects to one or more host computers over buses 36 at host processor ports 37 of the host adaptors. The host processors also can communicate with each other, for example over an Ethernet based bus 50 (Fig. 1).

Referring now to Figure 3, in general operation, a series of arguments or parameters describing the tests or test to be performed within the mass storage system is entered into a master host processor (step 60). The parameters, represented by the data entered into the master host processor, will define and effectively control the operations by which the hosts gather statistics describing performance of the mass storage system. The arguments or parameter data are entered into a main control program (step 62). Once the parameters are set in the main control program, operation transfers to a main driver program, running on the master host computer. The driver program controls operation not only of the master host computer, but of all of the other (client) host computers as well.

The driver program effects time synchronization of the host computers and causes the next lower level controller programs (the programs which control the commands and data sent to and received from the disk controller), to operate in time synchronization (step 64).

In order to achieve both time synchronization of the host computers, and accurate and timely operation of the mass storage system, the driver program may first cause each of the client host computers to synchronize its clock with the master host computer. Further, in response to a communication from the master computer, all host computers begin to issue commands to the controller of the mass storage system, based upon the arguments or parameters previously stored in their memories.

After the necessary test-defining parameters are stored in the memory of each respective host computer, the test is ready to proceed. The next lower level controller program in each of the host computers, designated the scripting program, causes each of the host computers to command the controller in accordance with the command information provided to it by the master host computer driver program (step 66). (Note that the master host computer itself also "receives" such information from the driver program.)

Depending upon the particular disk controller system, one of two possible methods of operation can proceed. If the controller is a "simple" controller, each host computer will itself measure and collect statistics (the raw data) identifying, for example, the response time for each command which it sends to the controller. These response times are collected in a manner which allows the host computer to identify to which command the response time corresponds. Alternatively, for a controller such as the EMC Symmetrix® controller, the controller itself can provide or supplement the raw data for each of the commands which it receives from the hosts. Under this latter circumstance, the controller will return not only the data requested by the command, but in addition, in response to special host requests, the statistics describing the response times for the commands which are being received. That information is provided to the particular host which requested the operation.

Each host computer, then, analyzes the response time data which it either has received or generated itself (step 68). In the illustrated embodiment of the invention, this raw data, which can amount to several gigabytes of information, is analyzed preferably, at each host computer. The analysis is basically a data reduction analysis whereby each host computer, rather than maintaining the full raw data set, operates to reduce the received response times to a smaller set of data. In a first particular embodiment, the data is placed into "buckets", each bucket representing, for example, 0.25 seconds. (In other embodiments,

differently sized buckets can be employed). The buckets, however, collectively represent a non-overlapping, continuous sequence of time durations.

In another embodiment of the invention, the response times can be accumulated for a period of time, so that the data returned to the master host computer will represent the cumulative response times for all commands issued during each of a plurality of non-overlapping contiguous larger durations of time, for example 5 or 10 seconds. That is, for each of the contiguous time periods, the response times for each command initiated in the period will be accumulated. This is particularly useful where the tests can run for several hours and in which tens of gigabytes of data will be produced for each of the host computers.

No matter what method is used to collect and/or reduce the data, the master host computer collects the resulting data from each other host computer (step 70). The master host computer, at the driver program level, then can further analyze the reduced data, as described below, to obtain and present statistics to the user (step 72). These additional statistics can provide further insight and understanding into the performance and operation of portions of or the entire of computer/memory system.

Referring now to the operation of the computer system in more detail, and referring to Figure 4, at the main program level, in the master host computer, a number of parameters or arguments are entered and recorded using a graphic user interface. These are illustrated in the table of Figure 4. Turning to the table, the initial parameters include the number of logical disks to be tested, the number of "child" processes to start (as that term is used in the Unix operating system), the number of processes that capture response times, the number of response times to collect, the buffer size requested, and the offset size, in bytes, rounded down from a randomly generated number (this supports seeks on random reads and writes to even boundaries of stripes). Other required arguments include the maximum range in megabytes to span the device, the time in seconds to effect read or write operations or the amount of data in actual bytes to read and write, and the percent of operations which will be read operations (with the remainder being write operations). Other optional arguments, in the illustrated embodiment, include identification of the devices to test, identification of which host will be the master host computer and whether the I/O operations will be sequential or random. Other optional arguments include the number of sequential I/O operations to perform, once the system has "seeked" to the correct offset for a random

operation, and the displacement in bytes back from that particular offset. In this particular embodiment of the invention, there are the yet further optional arguments which include the amount of time to delay between I/O commands, the initial byte offset to start sequential read or write commands, the method in which response start times will be collected (for example the use of buckets), a parameter identifying a percent hit rate to be implemented in connection with Integrated Cache Disk Arrays (ICDA's), including Intelligent Storage Systems, with controller cache to read or write a specific number of megabytes of data, and a random range multiplier (for devices larger than the scope of the random number generator).

Referring now to Figure 5, the operation of the system, in accordance with the invention, can be viewed as a series of nested loops, the outer most loop being the main program, the next loop being the driver program, and the inner loop being the scripting program. In the outer loop, the system receives the arguments or parameters which control or set up the operation of the test program. Those parameters or arguments have been described above in connection with the table of Figure. 4. Referring to Figure 5, the main program receives (at step 100) and enters (at step 102) the various arguments in its data files. In a preferred embodiment of the invention, each test is performed typically three times (tested at step 104) to ensure a statistical averaging which creates both confidence and accuracy, thereby avoiding variability and statistical anomalies. A test may also be performed once for verification purposes (and with less data).

Once the arguments have been stored in the data files on that host computer, which is designated as the master computer, the main program invokes the driver program, running on the master computer, to set up all of the hosts. This is indicated at step 106. The driver program, in response to the arguments provided, will initialize the cache memory, if necessary, and will initialize as well, all of the host computers, including the master host computer. The driver program, at 108, causes all of the host computers to be time synchronized. (If time synchronized, it may perform this function by sending to each of the client host computers over channel 50, the clock time of the master host computer.) Each client host computer now runs the scripting program in response to initialization by the driver program and communications from that program over communications channel 50, linking all of the host computers. (Thus all of the client host computer clocks may then be set in

some embodiments of the invention, with the result that all of the host computers participating in the test are operating with time synchronized clocks.)

The driver program then transfers to each of the client host computers the necessary configuration and parameter files with which the scripting program, at the client host computers and at the master host computer, will operate to test the mass storage system. This is indicated at step 110.

Next, the driver program initiates testing of the mass storage system by communicating to each host computer, directly from the master host computer and over the interconnecting communications channel 50. As a result, each host computer begins sending commands and data to and receives at least data from the mass storage system at the same time. At this point, it is the configuration and parameter input to the master host computer, as delivered to the client computers, which controls the actions of each of the client host computers. Thus, the provided information and arguments can cause, for example, only a subset of the host computers to communicate and issue commands to the mass storage system, and/or only a specific set of logical units at the mass storage level may be exercised in a specific configuration dictated by the arguments input at step 100.

The scripting program, when a test is complete, as tested at 114, then reduces the data which it has collected (step 116). As noted above, the data reduction process, if one is used, can use either a bucket compression or an accumulation compression approach. (The approach can be dynamically changed during the test sequence by, for example, a user command to the master host computer.) Alternatively, the initial raw data may be maintained in its original form. The raw data can include, for example, the response times to read and/or write operations, which have been commanded in accordance with the input parameters. (In accordance with the invention, when there is a mix of read and write commands, the system first issues a block of one set of the commands and then a block of the other set of commands. For example, when there are to be 40% read commands, the system can issue three write commands, followed by two read commands, followed by three write commands, etc. In this manner, the associated statistical data which is collected can be directly correlated to a particular read or write command.)

Once the data is in its final (and most likely, reduced) form at the client host computers, it is transferred over channel 50 to the master host computer. This is indicated at

step 118. Thereafter the master host computer can effect a more comprehensive data analysis, at step 120, to determine the performance of the mass storage system, for example, the number of I/O's, and in particular, the number of writes, as a function of time. The driver program then determines whether another test is to be performed, at step 104, and if not, the driver checks to determine whether further analysis of the data is to be performed (at step 122). If further analysis data is to be collected using a new configuration, the new configuration is generated at step 124, and the process begins again starting with step 110. If no further analysis data is needed, the system returns to the beginning of the process as indicated by loop 126. In this manner, the three nested loops of this preferred particular embodiment, loops represented by the main program, the driver program, and the scripting program, can provide effective and dynamic testing of the storage system to determine its performance under a large variety of situations.

Referring to Figure 6, in accordance with the operation of the system, there are provided, generally, six operational subsystems. These are designated administrative 200, data transfer 210, processing of raw data 220, post processing of raw data 230, reporting 240, and database and trend analysis 250. These elements of operation are initiated, by the user, in a sequential manner as noted in Figure 7, after the system for collecting the raw data (on the disk controller) is initiated at 260.

The data reduction workbench system of the invention begins with a "project tab" which creates or opens a directory structure to allow for management of the project data. This function organizes the data reduction workbench and allows creation and management of new, or the opening of existing, projects and test phases, and further insures consistent naming of all of the components within a project. It also logs all operations performed at both a project and a test phase level. At the test phase level, the system creates or opens the folders for a particular test phase which will hold all of the test data and informational data regarding that particular test. Each test is identified, at 290, and the organization process to effect data transfer begins. Next, in the data transfer phase, at 300, a file transfer protocol is incorporated for transferring the data from the test environment (either the host computers or the disk controller system, such as the EMC Symmetrix® controller system). This mechanism provides the directory structure of all test data and information data for the particular test (benchmark) which has been run.

Once the data has been transferred and is ready to be processed, the system processes the raw data, at 310, by reading the test files and creating summaries and error files for the test. The processing of raw data further creates statistical analysis program files, for example, for further processing and a cache ratio report to ensure that the actual cache ratio is within an acceptable margin of the targeted cache ratio for the characterization workload test. The data is also checked for errors at this stage and those errors are reported, and in some instances, corrected. The coalesced data is then formatted for input to the post processing stage at 320. The coalesced data results from processing the raw data, by automatically processing characterization and database simulation benchmark data, as well as controller generated internal data, and coalescing, according to the test, test iteration, and test configuration, the data collected by each host operating in the test. The coalesced data is thus provided in a single data file for the following post processing.

The post processing phase 320 provides an interface to the statistical analysis data processing program, and to other analysis tools such as the Write Analysis Tool described in U.S. application Ser. No. 09/542,463 entitled BENCHMARK TOOL WRITE ANAYSIS FOR A MASS STORAGE SYSTEM, and filed April 4, 2000, and incorporated herein in its entirety by reference, and allows for plot generation, data summarization, and the importing of the test and information data for a particular test, using a graphical user interface. As a result, formal plot presentations can be automatically loaded into an electronic document, for presentation in a standardized format.

Once post processing has been completed, the system provides interfaces to other presentation formats to generate spreadsheets from the statistical analysis program summary files created during post processing to enable charts and tables to be created in, for example, a standardized word processing report. This enables the development of performance reports. This is indicated at 330. In addition to the reports, this system also automatically loads the benchmark statistical analysis summary data into a database (at 340). With the data thus available, a trend analysis can be created and displayed according to the test types and system configurations. This is indicated at 350. This enables the comparison of performance behaviors across multiple projects (that is, as a function of time) and test phases (for example as a result of different operating software for the same configuration of equipment) in order

to determine visually, at first impression, trends which may affect the analysis of system performance.

Post Processing

Once the data is collected, according to the invention, the system organizes, validates, checks, and presents the data as noted below. In the raw data post processing and correction stage, the validity checks can include checking individual data points for errors and applying predetermined decision criteria to either correct, ignore, or delete erroneous data. In addition, an overall validity comparison of the actual (and corrected) results with expected results based on the I/O configuration profile can be effected. The user of pre-established criteria to decide whether to correct, ignore or use the erroneous data is implemented in accordance with one embodiment of the invention by flagging any inconsistencies based on expected results of a particular input/output type and storing such flagged data in a system data files. Once the corrections have been effected, the revised data can be put into any of a number of industry standard templates.

Referring to Figure 9, a post processing "tab" enables the user to first start a statistical analysis program by clicking on button 900. The user can then bring up the statistical analysis program window to watch, on screen, for errors and to use it during any update-objects-routine which may follow. The user then selects the number of graphs per page as indicated at 902 (in Figure 8A, 8B, ... four graphs per page have been selected) and thereafter can select processing of the data as described above.

Accordingly, during the post processing phase of operation, in a preferred embodiment of the invention, referring to Figures 9, 10A and 10B, the system first checks individual data points against predetermined criteria to determine if any errors have occurred. Any error data are then, in accordance with the settings by a user in the graphical user interface, either corrected, ignored, or deleted. This is indicated at 1002. The results are then compared with the expected results for this particular I/O configuration profile. This is indicated at 1004. Any deviations from standard expectations are flagged at 1006. The results then are placed into a data files in a format for later presentation and operation.

The various data files are summarized in Figure 11. These data files are typically statistical analysis program related files and can be used to create database simulation objects as well as being placed into the correct report format as noted above.

Trends Analysis

The trend analysis referred to above, taking place as indicated at 350, provides a series of graphs, such as those illustrated in Figures 8A, 8B,.... These graphs provide a display of the collected data in a format easily viewable by the user so as to enable the user to understand and recognize trends in the data as a function of changes in one or more parameters. Thus, changes which result over time can be plotted to clearly enable the user to see and analyze, for example, the number of I/O's per second as a result of changes in the parameter, or the number of megabits per second throughput by the controller. The illustrative presentations of Figures 8A, 8B,... enable such presentations to be made effectively, and preferably in color.

Referring more particularly to Figure 12, which is a screen shot useful in connection with the trends analysis, a user can browse the available databases by selecting the "Select DB" button 1200 and selecting a database of interest. Thereafter, the system will load the database information into the form (Figure 12) creating a selection tree 1202 on the left side, as shown, and all of the default test types and configurations will be provided in the drop down boxes.

If the chart selection was "on-line", the analysis generates the chart on the display screen. The chart can also, as an alternative, for example, be placed in a word processing document by selecting "test type" or "view all", for example, at the chart selection 1204.

The user selects the test phases which are to be graphed together. By default, the project and one test phase will be selected, however, all test phases for a project can be selected by clicking, for example, on the project. Alternatively, specific desired test phases can be selected (or deselected) as needed. As noted above, the graphs can be generated and viewed online, or printed, and can be shown according to test type or types as required. The system also, by pressing the "scale" selection 1206, will calculate the minimum and maximum values for the data and use those values to scale the graph. A percent scale is also available where the largest value of the chart is considered at 100% and all of the values are displayed accordingly.

The various specific information is shown in the characterization frame 1208 and is selected by the user and viewed as a chart online or report.

During the input process, the user selects, using the graphical user interface, the parameters which enable the graphs of Figure 8 to be generated. The data corresponding to the graphs have been previously created in the database of the system. When the graphs are presented, as illustrated in Figures 8A, 8B,..., the presentation identifies the parameters relating to the graph including the ports of the controller which have been used, the nature of the tests, such as a random delayed fast write, the size of the blocks which have been used, and other test parameters as indicated in the Figures 8A, 8B,....

By plotting this information in a graphical format, the user is enabled to spot trends in the data as a result of changes over time, or other parameters. This data is also available for viewing on screen.

<u>Graphical User Interface</u>

The system provides for a graphical user interface which enables operational parameters of the system to be created quickly, with repeatability, reliability, and use by a much broader audience. The graphic user interface is essentially a front-end device for the invention which automatically operates to generate and/or work on three types of files: configuration rule data, workload data, and benchmark data. It allows the selection of various test types based upon the user inputs which are provided in a "point and click" manner. The graphic user interface of the invention is substantially more reliable for the user and enables the user to quickly and easily define the system tests which are to be performed.

The typical graphic user interface is presented, in accordance with the invention, in Figures 9A, 9B,.... In accordance with these Figures, and referring to the nomenclature either well known in the art or noted above in connection with various adaptor elements of a system, the various elements of the interface are indicated and described therein.

The user is thus enabled to form this presentation using the graphical user interface.

Additions, subtractions, and other modifications of the illustrated embodiment of the invention will be apparent to those practicing in this field and are within the scope of the following claims.

WHAT IS CLAIMED IS:

1. A method for measuring system performance in a mass storage system, the storage system having a plurality of disk drive storage elements controlled by a disk drive controller, said controller receiving commands and data from and returning at least data to a plurality of host computers, said method comprising the steps of

enabling a graphical user interface for generating an input parameter containing sequence input of commands for operating said system for measuring system performance,

generating from said input parameter sequence a test sequence input identifying commands to be sent to the storage system,

executing at at least one host computer a test request identified by said test sequence input, by sending commands to said mass storage system,

accumulating, at at least said executing host computer, data regarding performance of said mass storage system, in response to the requests sent by said host computer, and

processing said accumulated data regarding the performance of said mass storage system in response at least to said one host-generated command

2. The method of claim 1 further comprising generating at said graphical user interface at least one of configuration data, workbench data, and benchmark data.

3. The method of claim 2 further comprising generating at said graphical user interface configuration data, workbench data, and benchmark data.

4. The method of claim 1 further comprising selecting, user the graphical user interface, from various test types for the input sequence of commands, in a point and click fashion.

5. The method of claim 4 wherein said test types include defining a system configuration, test periods, and sequence of test repeats.

6. The method of claim 1 further comprising identifying details of a storage system configuration and workload using the graphical user interface.

ABSTRACT

GRAPHICAL USER INPUT INTERFACE FOR TESTING PERFORMANCE OF A MASS STORAGE SYSTEM

A method for measuring mass storage system performance in which the mass storage system has a plurality of disk drive storage elements controlled by a disk drive controller, the controller typically having a cache memory, and the controller receiving commands and data from and returning at least data to a plurality of host computers, provides the flexibility of issuing commands to the controller in a variety of different configurations from a plurality of hosts in a time synchronized and organized fashion. Some significant data reduction techniques control and organize the data for later analysis according to the invention. Time delays and different configurations of both host and storage elements can be effected easily using a novel graphical user interface.

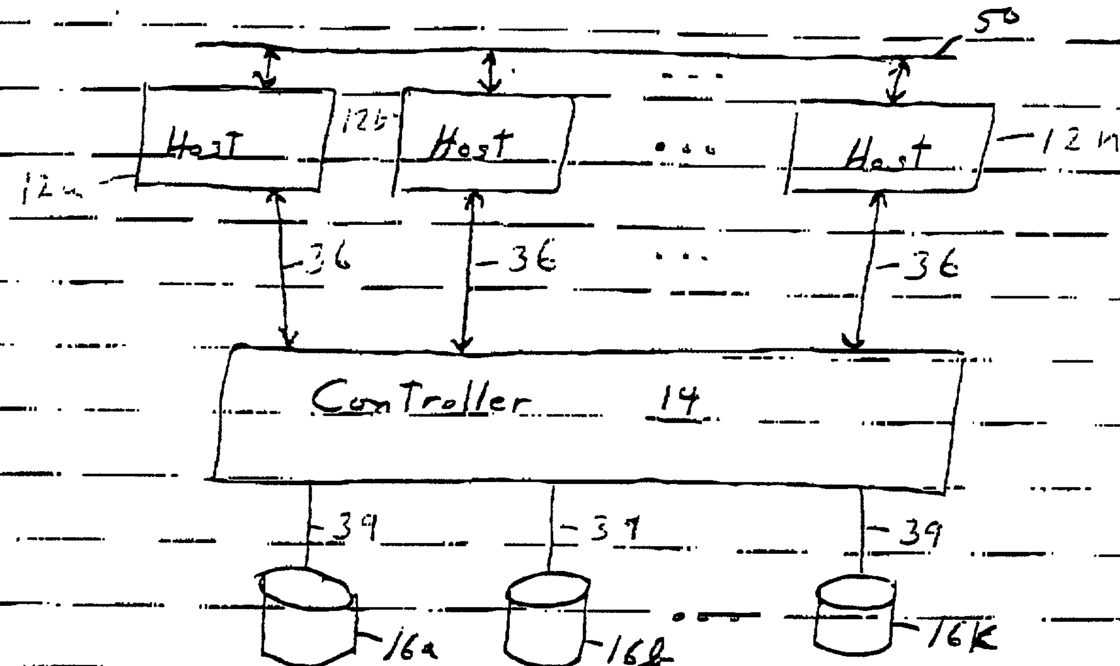

50
Host
12a
12b
Host
Host
12n
36
36
36
Controller 14
39
39
39
16a
16b
16K

Fig. 1

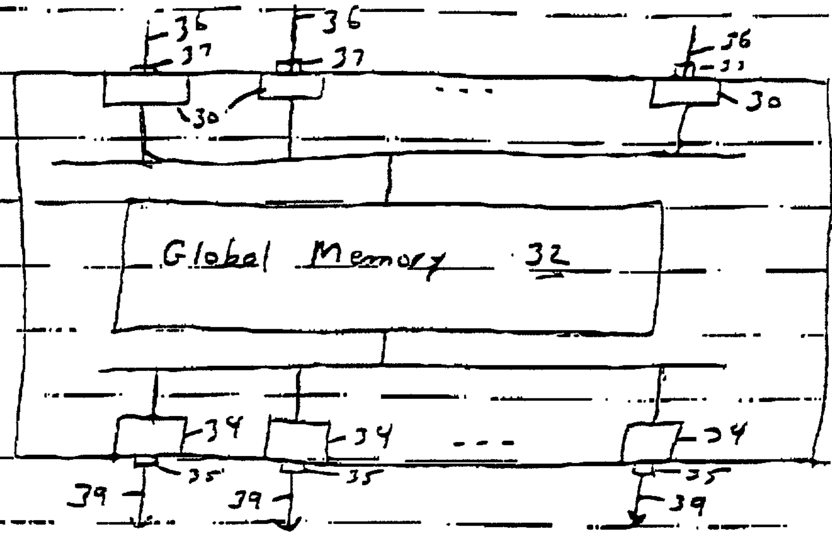


Fig. 2

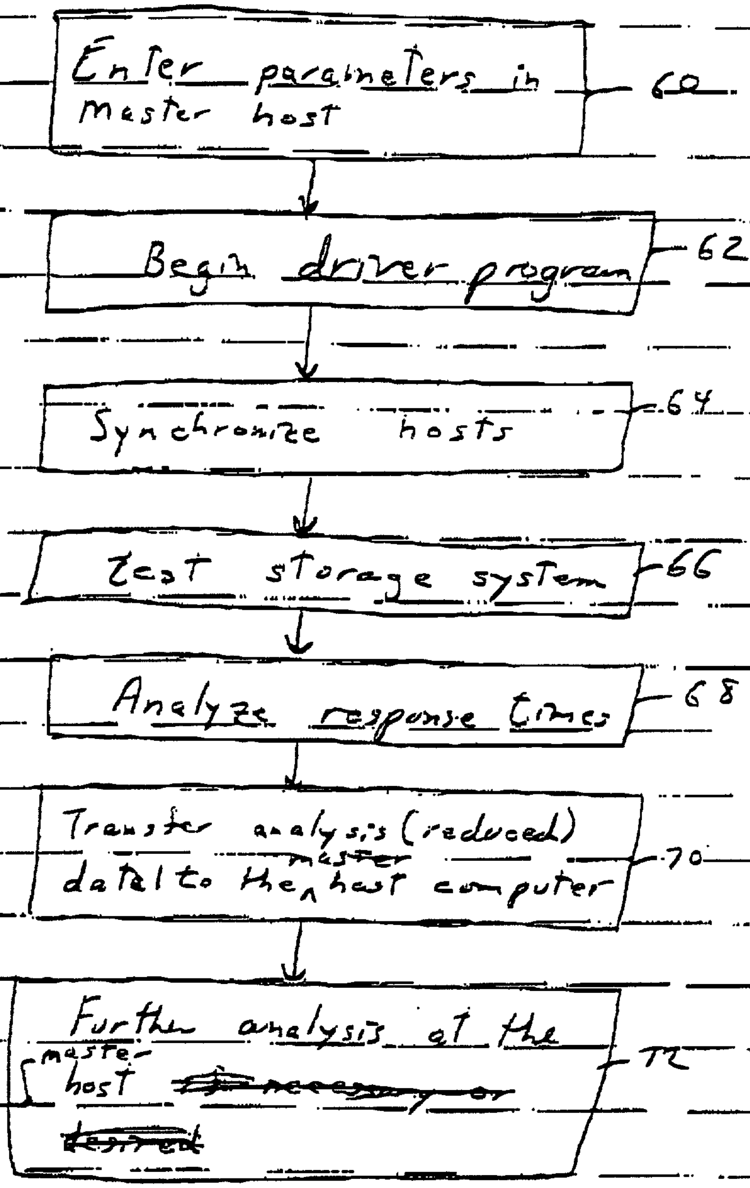

Enter parameters in master host
60
Begin driver program
62
Synchronize hosts
64
Test storage system
66
Analyze response times
68
Transfer analysis (reduced) data to the master host computer
70
Further analysis at the master host ~~if necessary or desired~~
72

Fig. 3

	Number of logical disks
	Number of "child" processes to start
	Number of capture response times
	Number of response times
REQUIRED	Buffer size
	Offset size
	Maximum range
	time of test
	read/write size
	read/write mix
	ID of devices being tested
	ID of master & client hosts
	I/O type (sequential or random)
	Number of I/O operations performed to correct ~~offset~~
	Displacement from offset
	Delay between commands
	Initial byte offset
OPTIONAL	Number of seeks for random I/O
	Data reduction method
	ICDA percent hit rate

Fig. 4

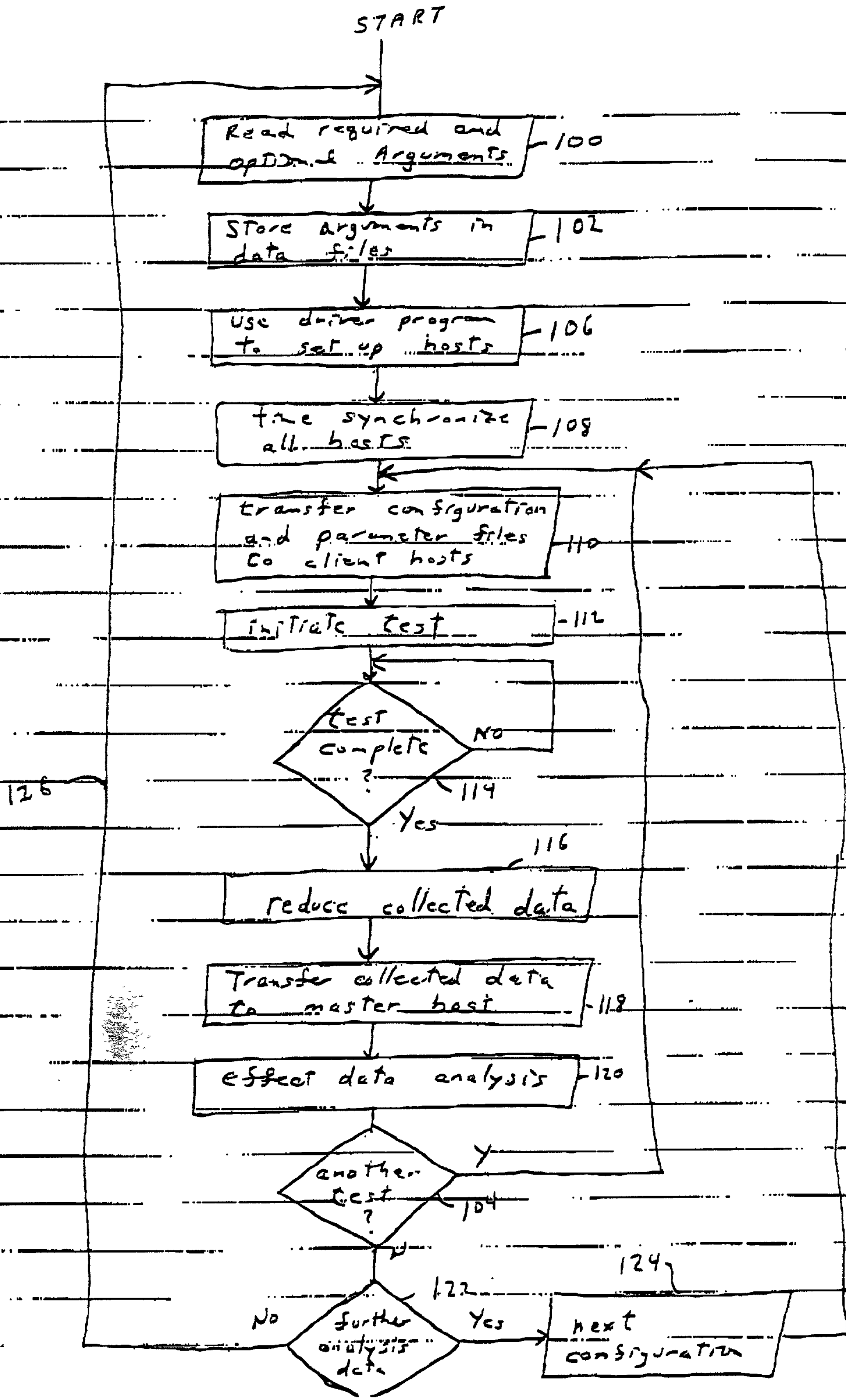

START
Read required and optional Arguments
100
Store Arguments in data files
102
Use driver program to set up hosts
106
time synchronize all hosts
108
transfer configuration and parameter files to client hosts
110
initiate test
112
test complete ?
No
Yes
114
126
116
reduce collected data
Transfer collected data to master host
118
effect data analysis
120
another test ?
Y
104
124
122
No
further analysis data
Yes
next configuration

Fig. 5

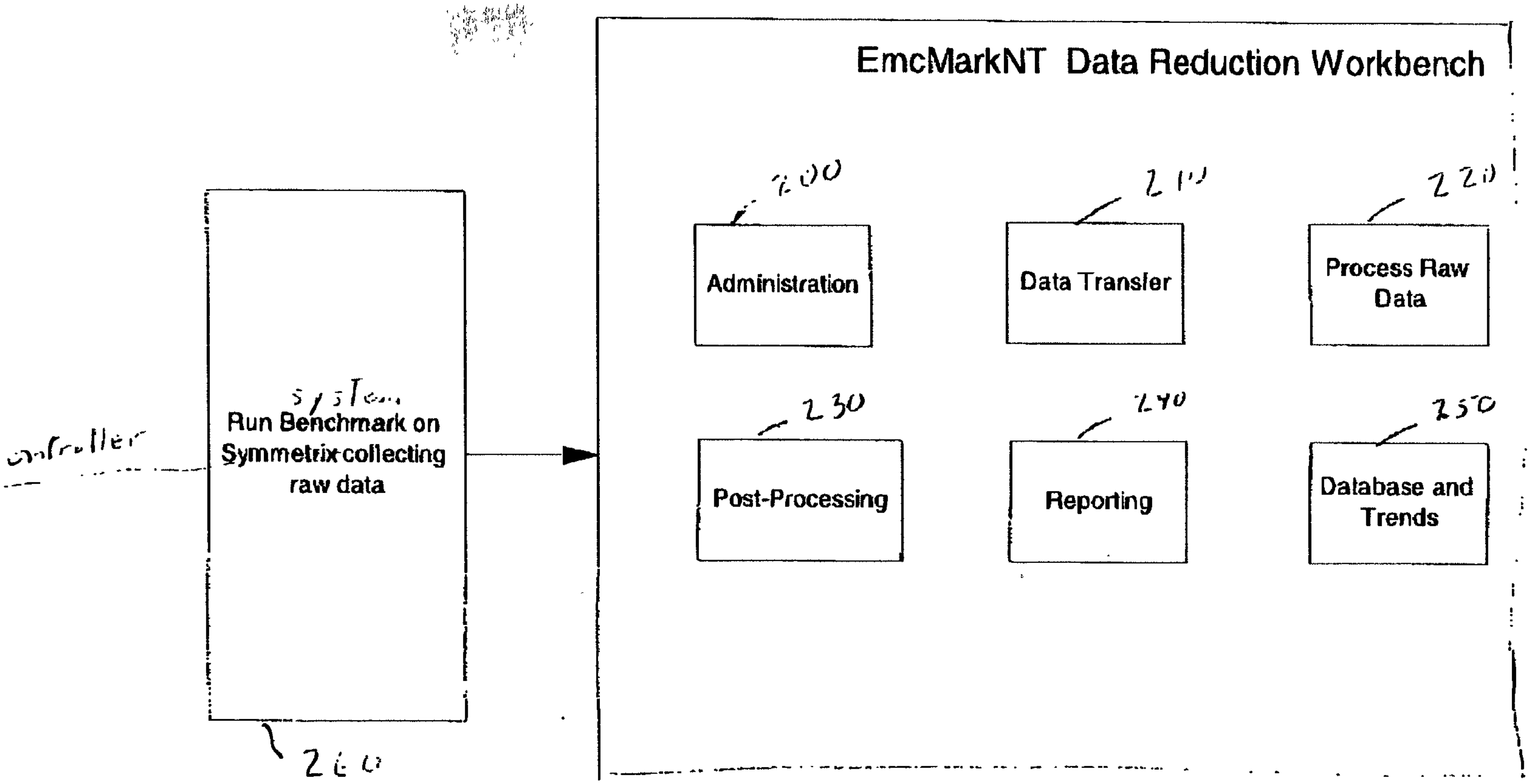

EmcMarkNT Data Reduction Workbench
200
Administration
210
Data Transfer
220
Process Raw Data
230
Post-Processing
240
Reporting
250
Database and Trends
controller
system
Run Benchmark on Symmetrix collecting raw data
260

Fig. 6

EmcMarkNT Data Reduction Workbench Flow

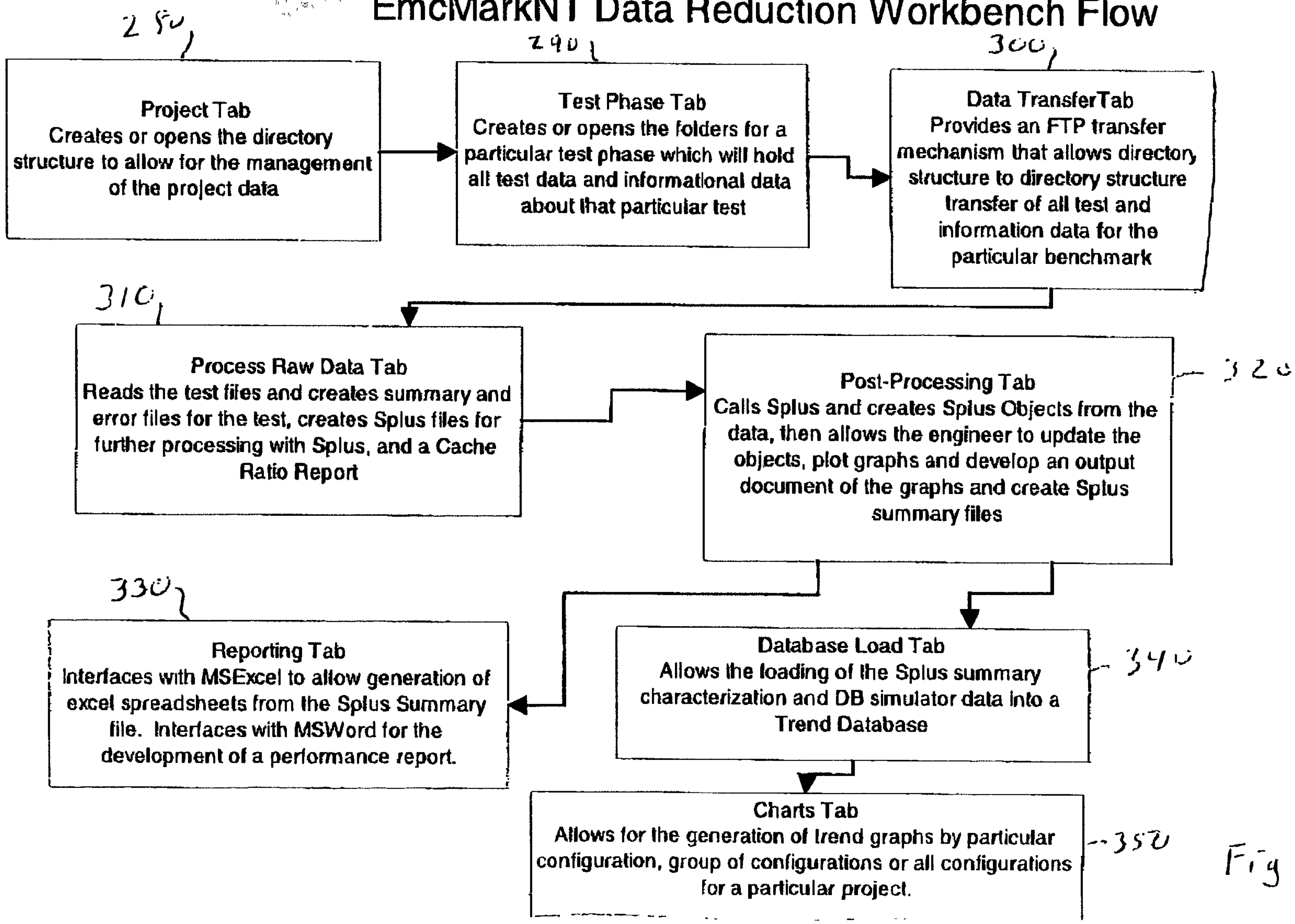
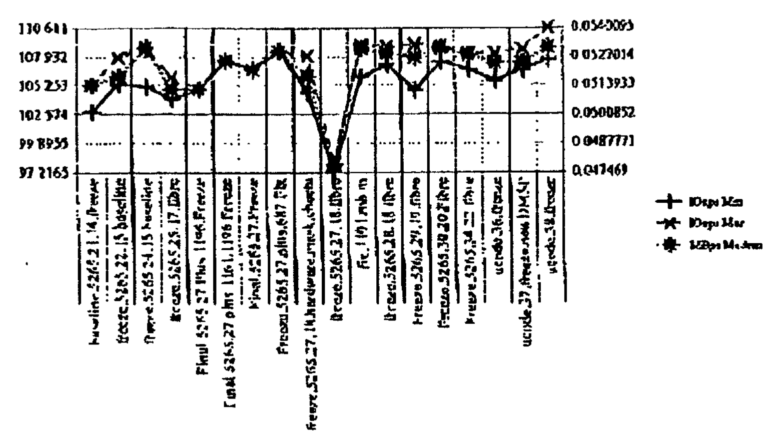


Fig. 7

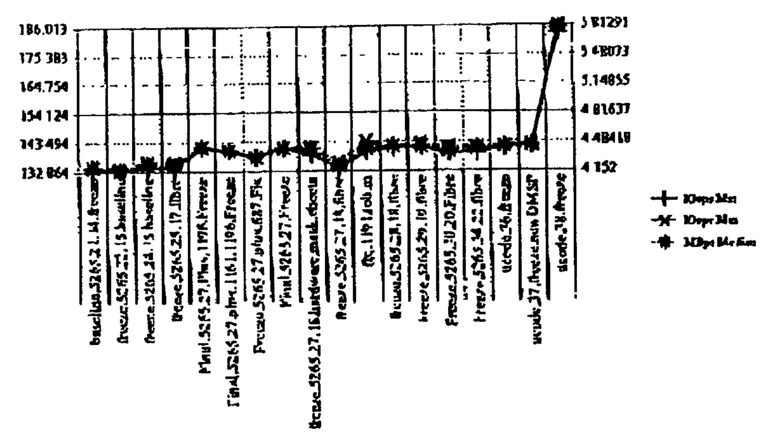
symm48.5265.fibre

IOsps and MBps for Random Delayed Fast Write - Req Size 512



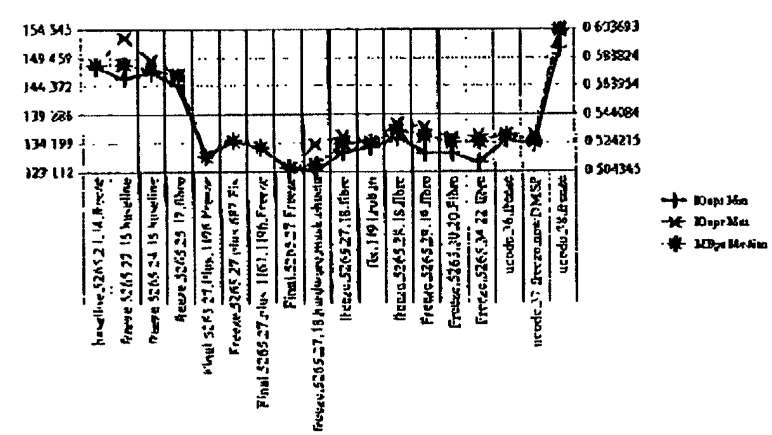
BA - 1 BAProc - 1 BAPort - 1 DA - 1 DAProc - 2 DAPort - 2 Drive - 1 Lun - 1 Hyper - 1

IOsps and MBps for Random Delayed Fast Write - Req Size 32768



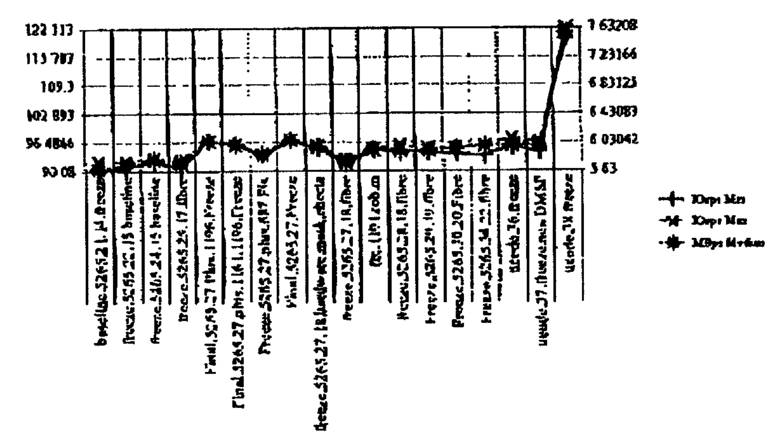
BA - 1 BAProc - 1 BAPort - 1 DA - 2 DAProc - 1 DAPort - 1 Drive - 1 Lun - 1 Hyper - 1

IOsps and MBps for Random Delayed Fast Write - Req Size 4096



BA - 1 BAProc - 1 BAPort 1 DA - 2 DAProc - 2 DAPort - 2 Drive - 1 Lun - 1 Hyper - 1

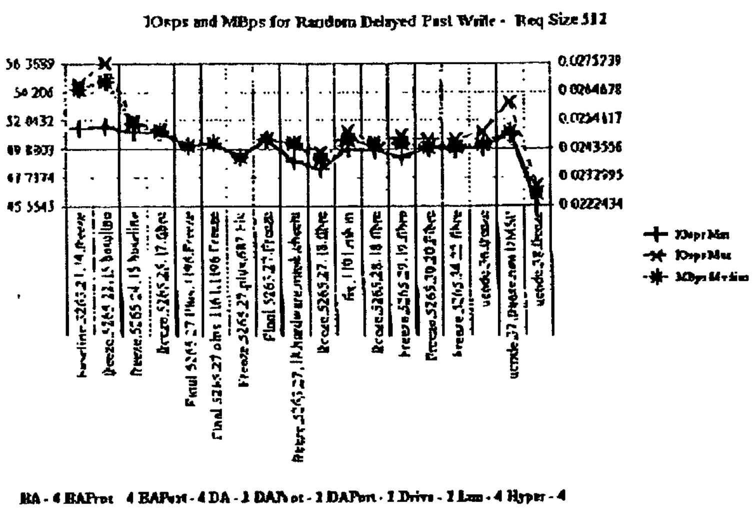
IOsps and MBps for Random Delayed Fast Write - Req Size 65536

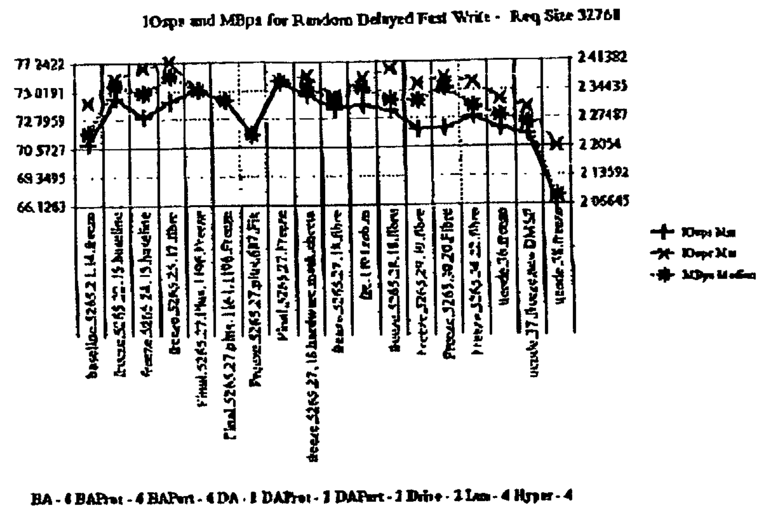


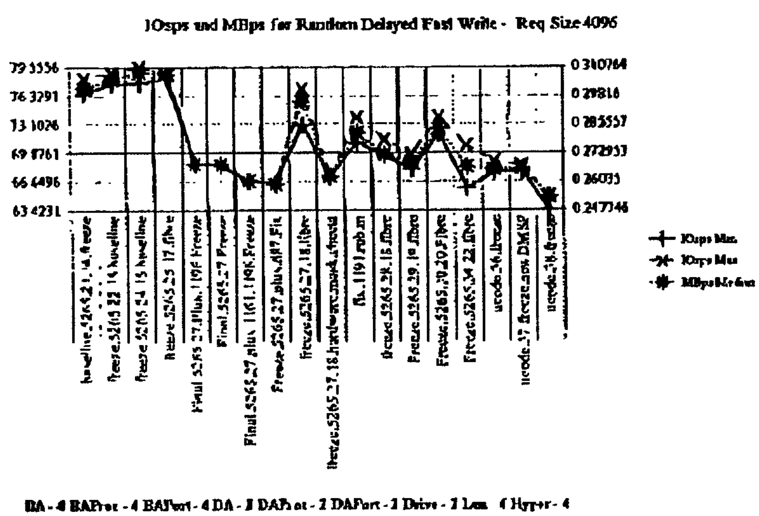
BA - 1 BAProc - 1 BAPort - 1 DA - 1 DAProc - 1 DAPort - 2 Drive - 1 Lun - 1 Hyper - 1

Fig 8A

symm48.5265.fibre







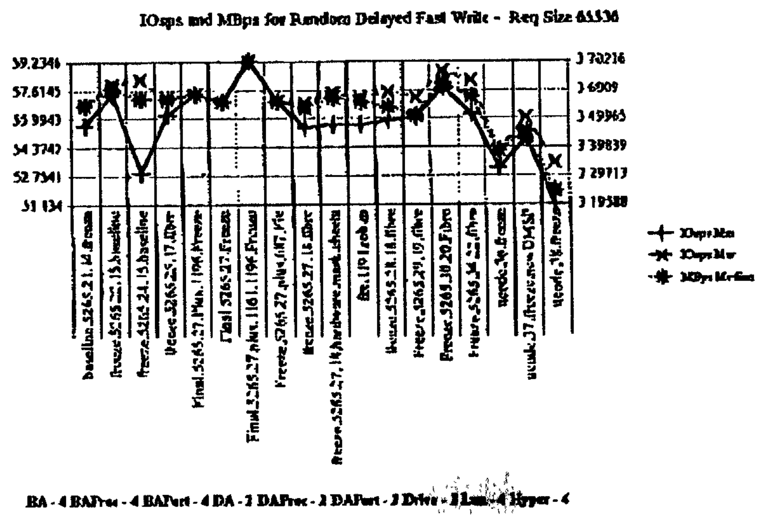
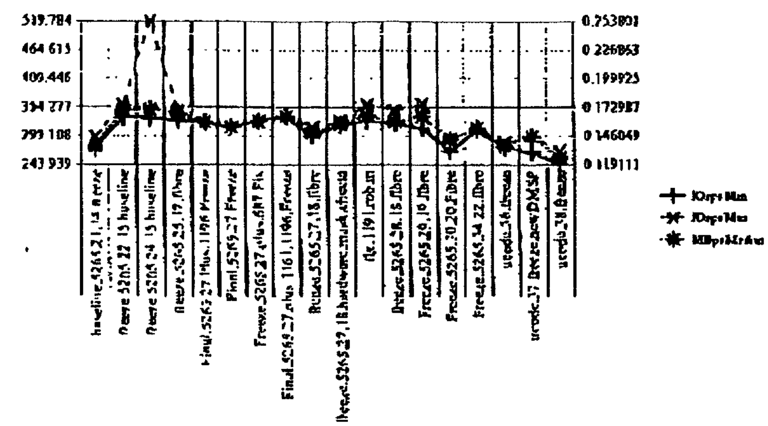


Fig 8B

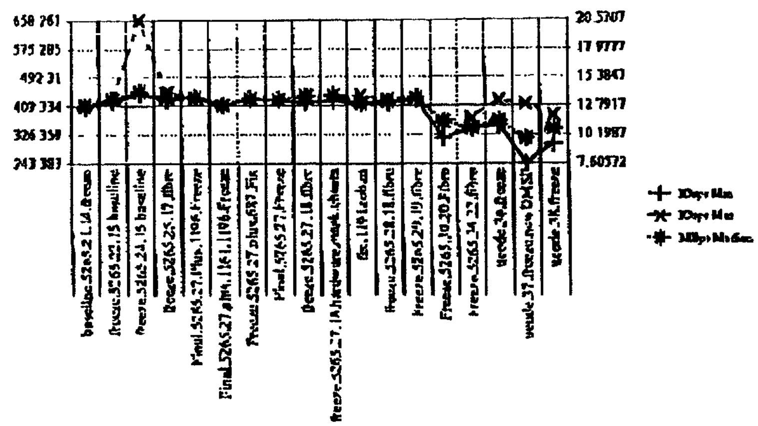
symm48.5265.fibre

IOsps and MBps for Random Delayed Fast Write - Req Size 512



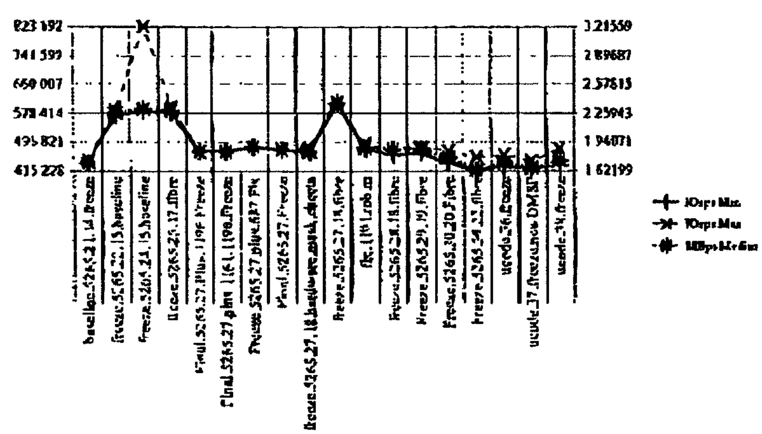
BA - 4 BAProt - 8 BAPort - 8 DA - 2 DAProt - 2 DAPort - 2 Drive - 12 Lun - 24 Hyper - 4

IOsps and MBps for Random Delayed Fast Write - Req Size 32768



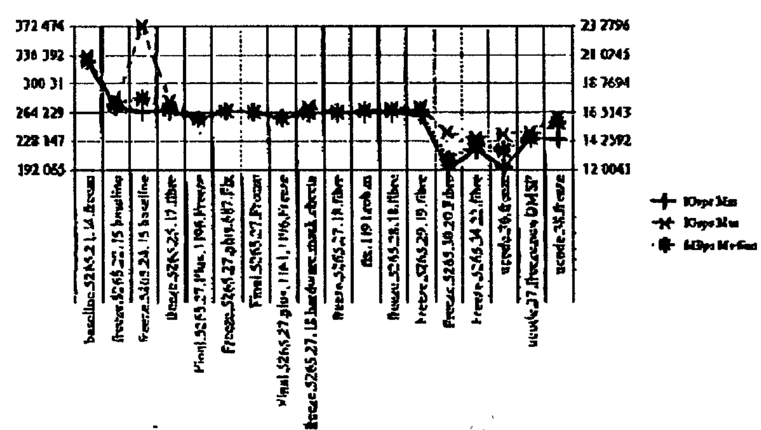
BA - 4 BAProt - 8 BAPort - 8 DA - 2 DAProt - 2 DAPort - 2 Drive - 12 Lun - 24 Hyper - 4

IOsps and MBps for Random Delayed Fast Write - Req Size 4096



BA - 4 BAProt - 8 BAPort - 8 DA - 2 DAProt - 2 DAPort - 2 Drive - 12 Lun - 24 Hyper - 4

IOsps and MBps for Random Delayed Fast Write - Req Size 65536

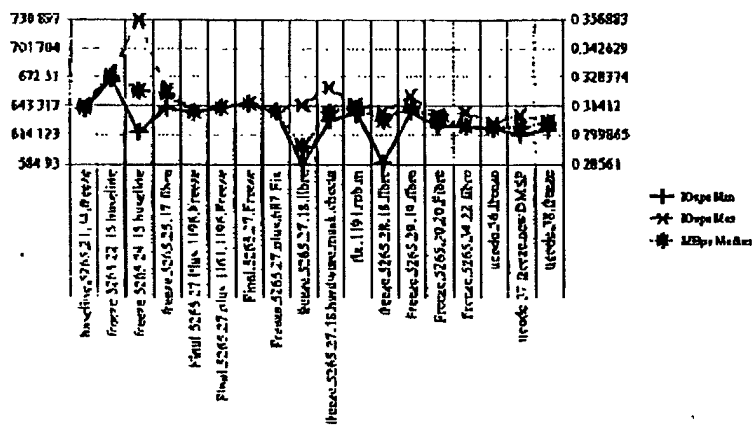


BA - 4 BAProt - 8 BAPort - 8 DA - 2 DAProt - 2 DAPort - 2 Drive - 12 Lun - 24 Hyper - 4

Fig 8C

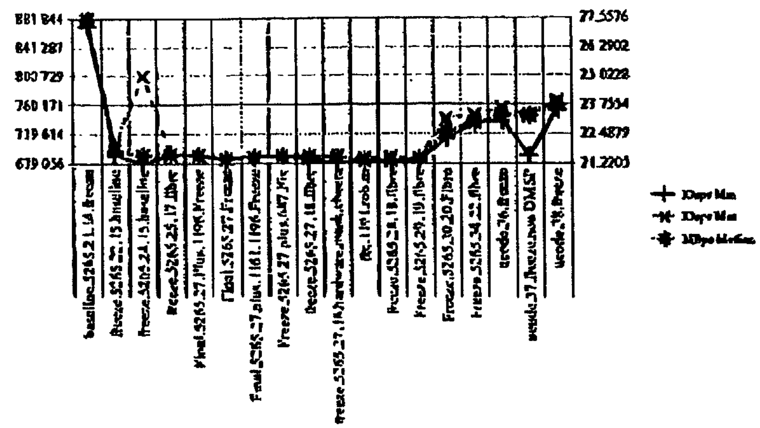
symm48.5265.fibre

IOsps and MBps for Random Delayed Fast Write - Req Size 512



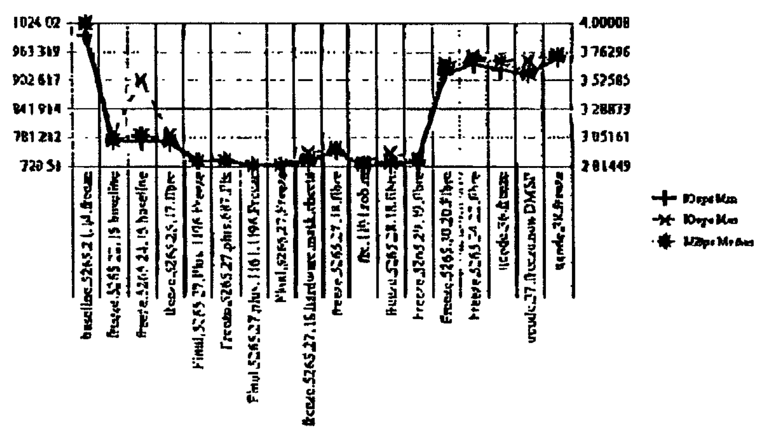
BA - 4 BAPrt - 8 BAPort - 8 DA - 2 DAPrt - 2 DAPort - 4 Drive - 24 Lun - 48 Hyper - 4

IOsps and MBps for Random Delayed Fast Write - Req Size 32768



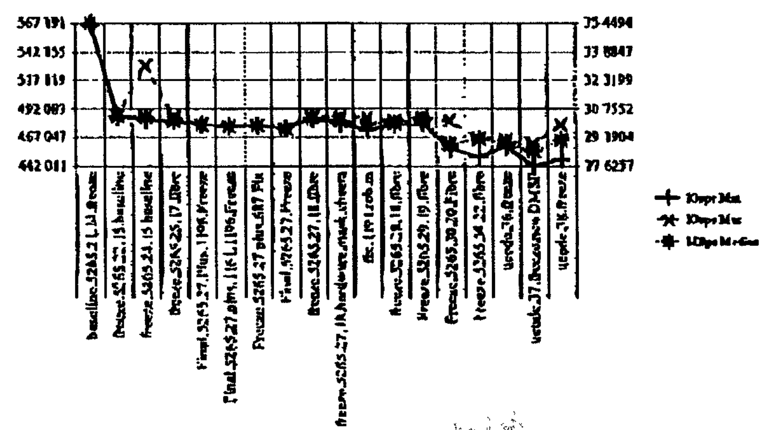
BA - 4 BAPrt - 8 BAPort - 8 DA - 2 DAPrt - 2 DAPort - 4 Drive - 24 Lun - 48 Hyper - 4

IOsps and MBps for Random Delayed Fast Write - Req Size 4096



BA - 4 BAPrt - 8 BAPort - 8 DA - 2 DAPrt - 2 DAPort - 4 Drive - 24 Lun - 48 Hyper - 4

IOsps and MBps for Random Delayed Fast Write - Req Size 65536



BA - 4 BAPrt - 8 BAPort - 8 DA - 2 DAPrt - 2 DAPort - 4 Drive - 24 Lun - 48 Hyper - 4

Fig 8P

Post Processing Tab

The Post Processing Tab creates objects, plot graphs and generates summary files using the Splus Data Analysis Software.

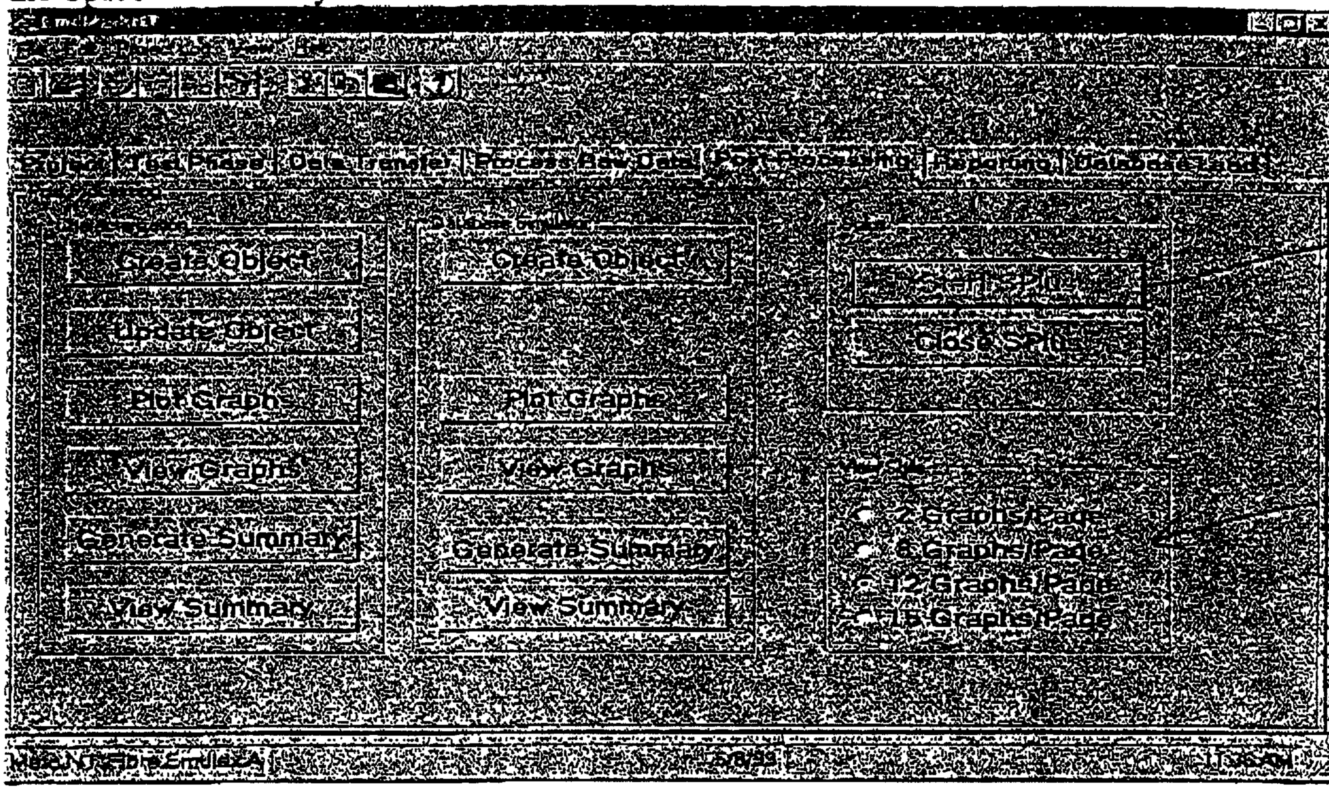


Fig. 4

Starting SPlus

1. Click on the Start Splus Button
   (Object buttons will be grayed out until Splus is running. If graphs or summary files have already been created those buttons will be visible)

2. Bring up the Splus window to watch for errors and to use during the Update Objects routine

3. Select 2 graphs/page 8 graphs/page, 12 graphs/page or 15 graphs/pageoption for viewing the graphs once generated

4. **Process Characterization or Database Simulator objects follow instructions on the next page.**

5. Select the Close Splus button when you are leaving this tab
   (If there is a problem closing Splus. bring up the window and close manually. Select NO twice to its Save Reports and Objects questions)

   (If you forget to close Splus before you exit the EMCMarkNT Data Reduction Tool you will need to quit out of Splus from the command line by typing **q()** or by file -> exit)

6. Go to the Reporting Tab

Symmetrix Configuration View

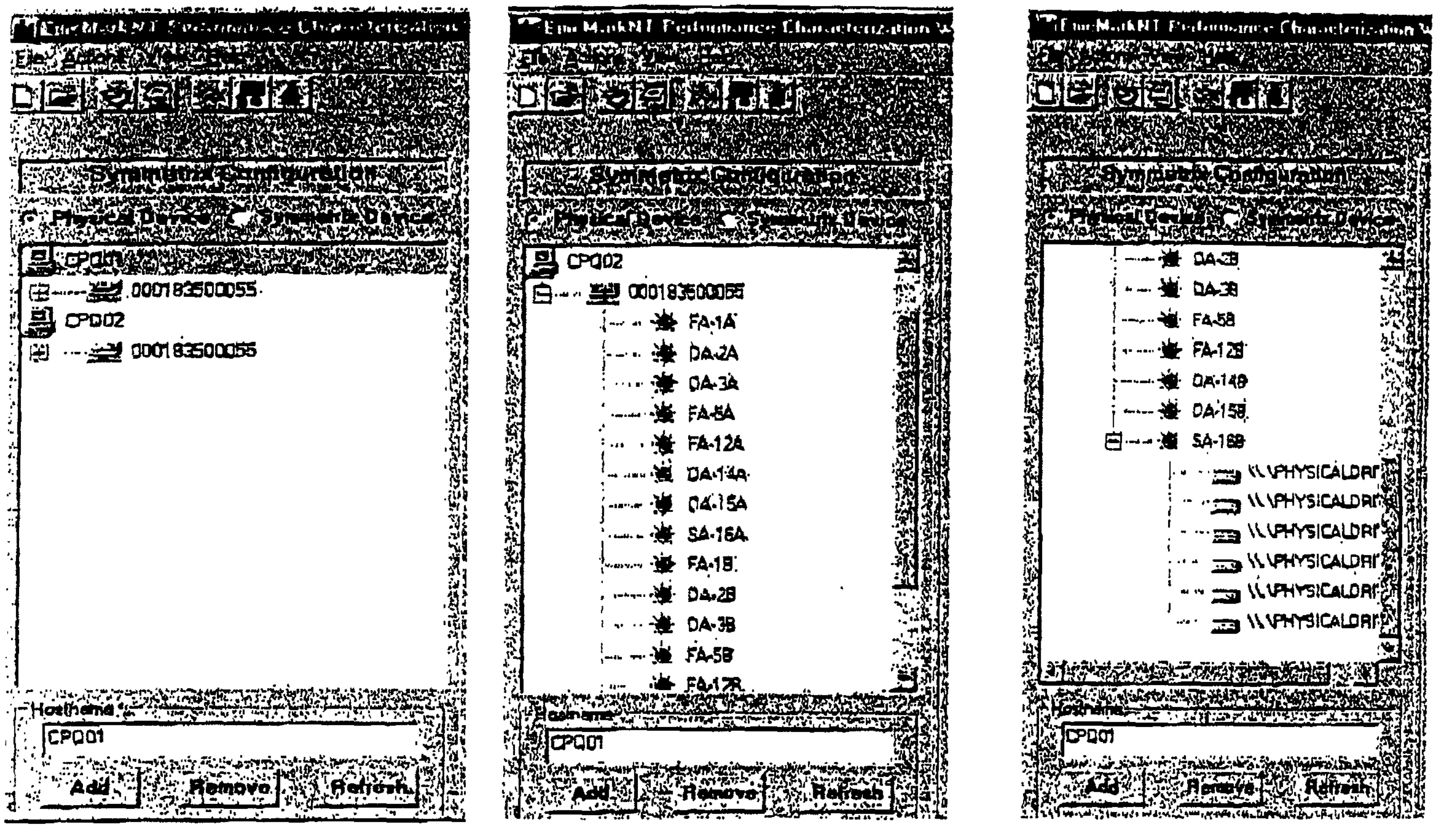


Fig. 9A

Lists the hosts and Symmetrix system
When the Symmetrix is expanded, BAs and DAs and will be displayed. Red indicates inactive and green indicates active

Physical Devices – will list the physical device names connected to the Symmetrix
Symmetrix Devices - will list the Symmetrix device names connected to the Symmetrix

Hostname – is the host highlighted on the list

The first host in the list is considered the Master Host
If no host is listed then the host you are on is considered the Master Host

Local, Remote, Gateway

If the Master Host is the host you are on then the job will run locally
If the Master Host is not the host you are then the job will run remotely, except
If there is a gateway setup in the Environment Tab, then the job will run through the gateway

Add – will add the host name typed in the Hostname box
Remove – will remove the host name typed in the Hostname box
Refresh – will refresh the host/Symmetrix information

Device Details

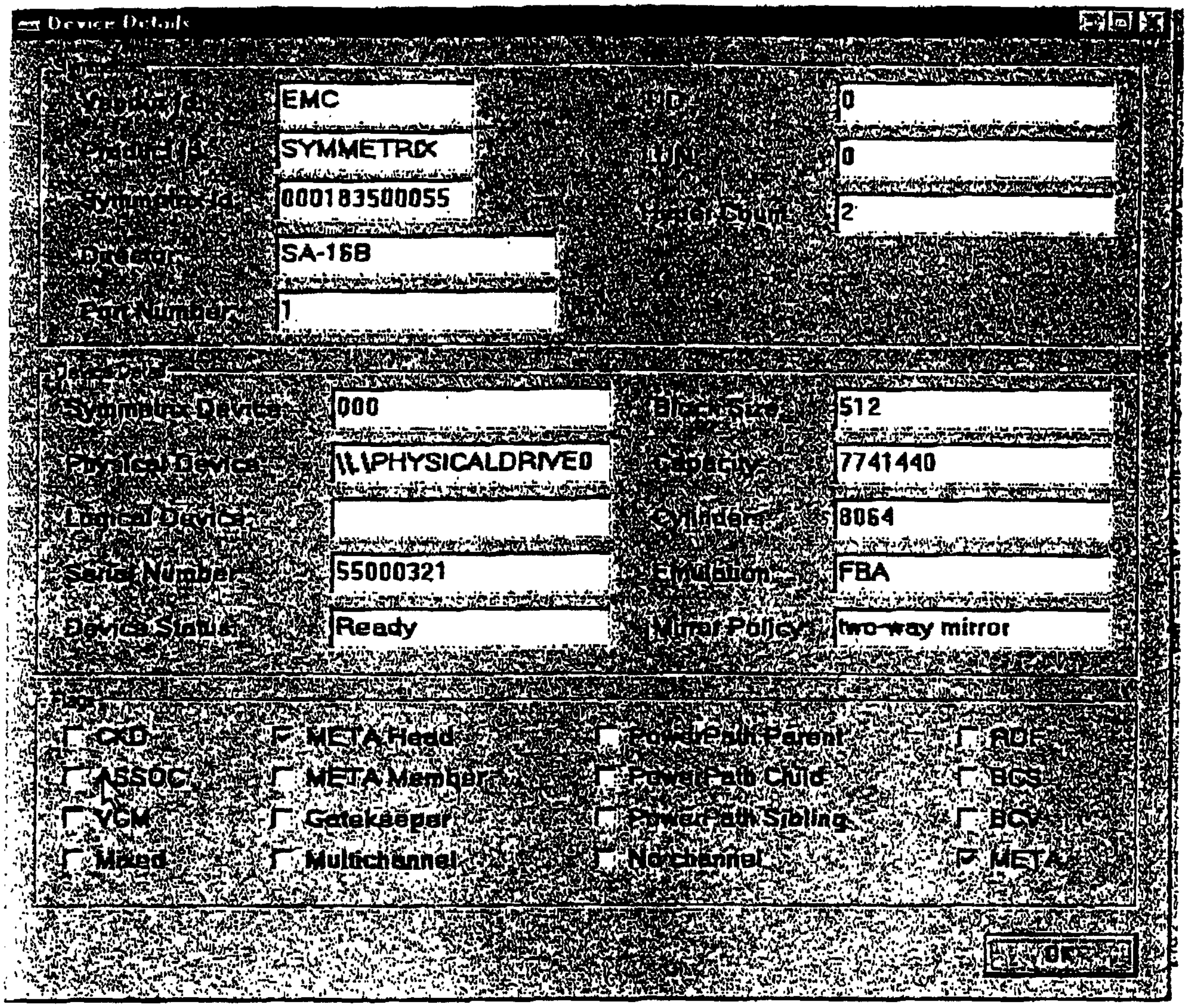


FIG. 9B

Symmetrix Details

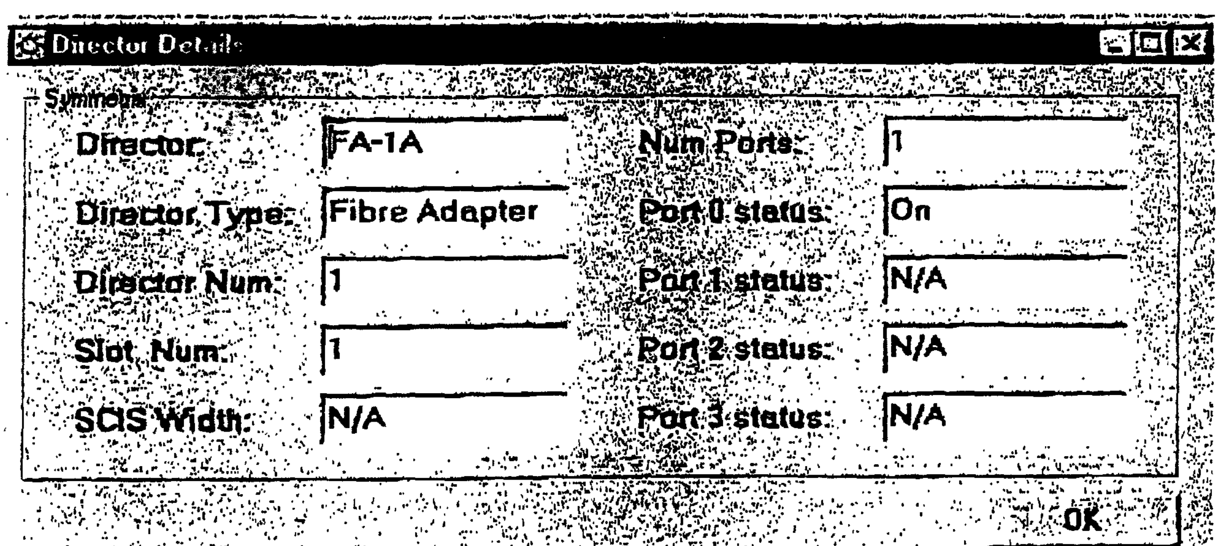


FIG. 9C

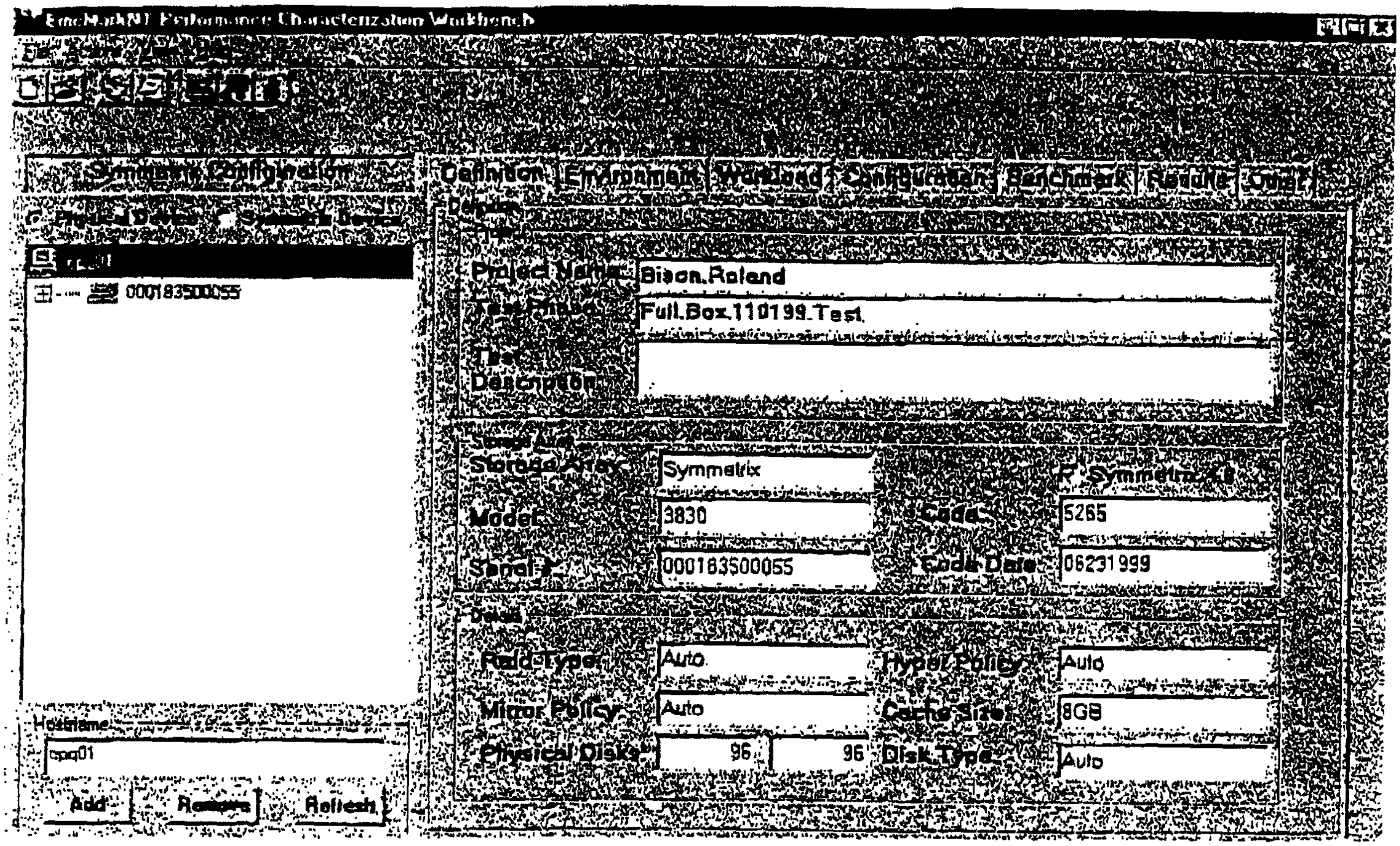


FIG. 9D

Project name – The project set up for this test.

- Select the New Project button to setup a new project, or the Open Project button to open a different Project.

Test Phase – the Test Phase setup for this test

- Select the New Test Phase button to setup a new test phase, or the Open Test Phase button to open a different Test phase under this project.

Test Description – comes from the ini file located in the Test Phase/Scripts folder

Storage Array Frame and Details Frame information from the ini file located in the Test Phase/Scripts folder
You can manually update the fields, or if you double click on the Symmetrix box the information gathered from the Symmetrix will be populated into those fields and upon exit will be written to the ini file

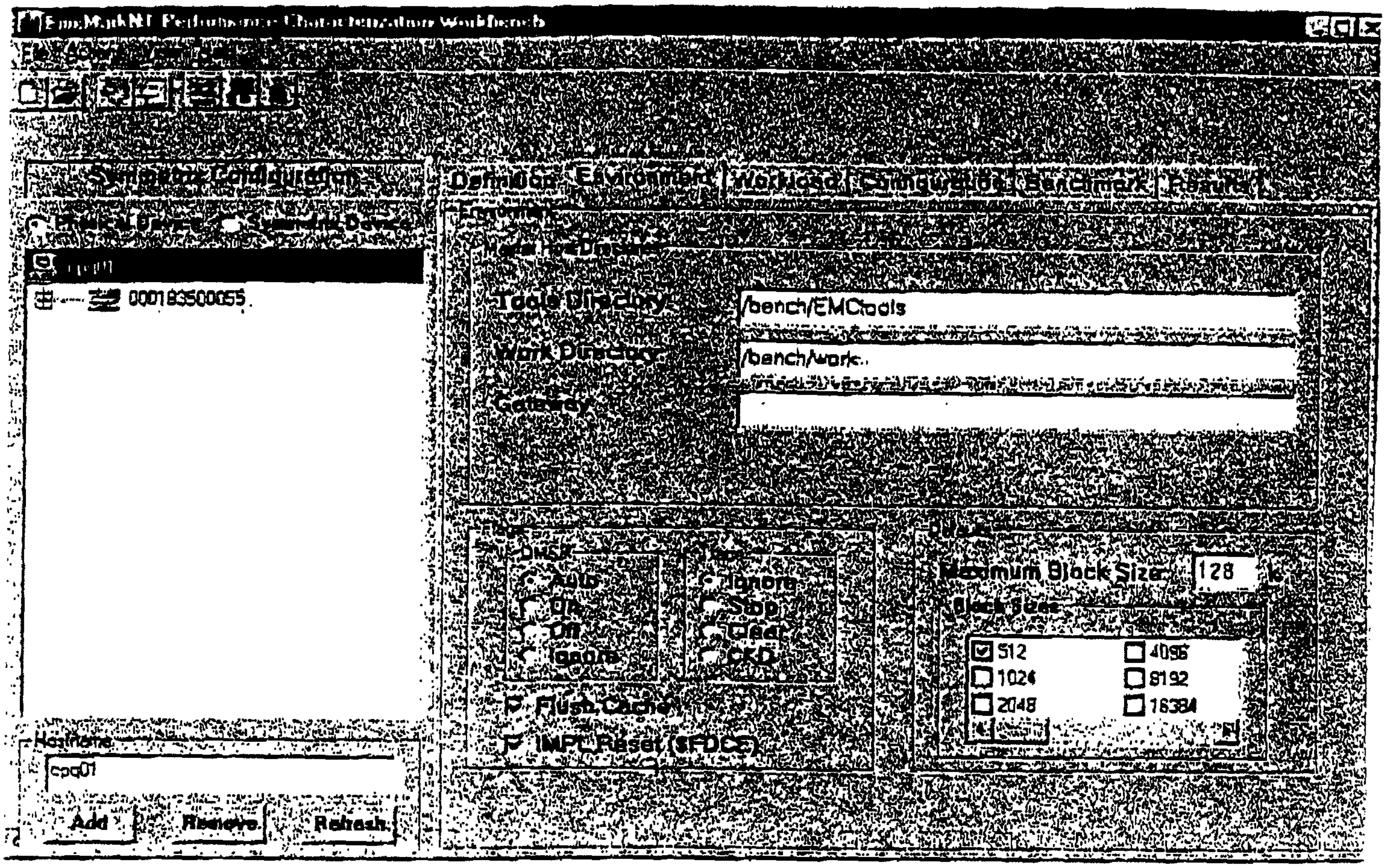


FIG 9E

Tools Directory – where the master scripts are located
Work Directory – your personal work folder
Flags -
DMSP
Trace
Flush Cache -
IMPL Reset ($FDCE) -
Defaults – Maximum Block size set to 128k
- Default blocks sizes selected for Workload when run directly from the Workbench

-not saved upon exit, must reset each time

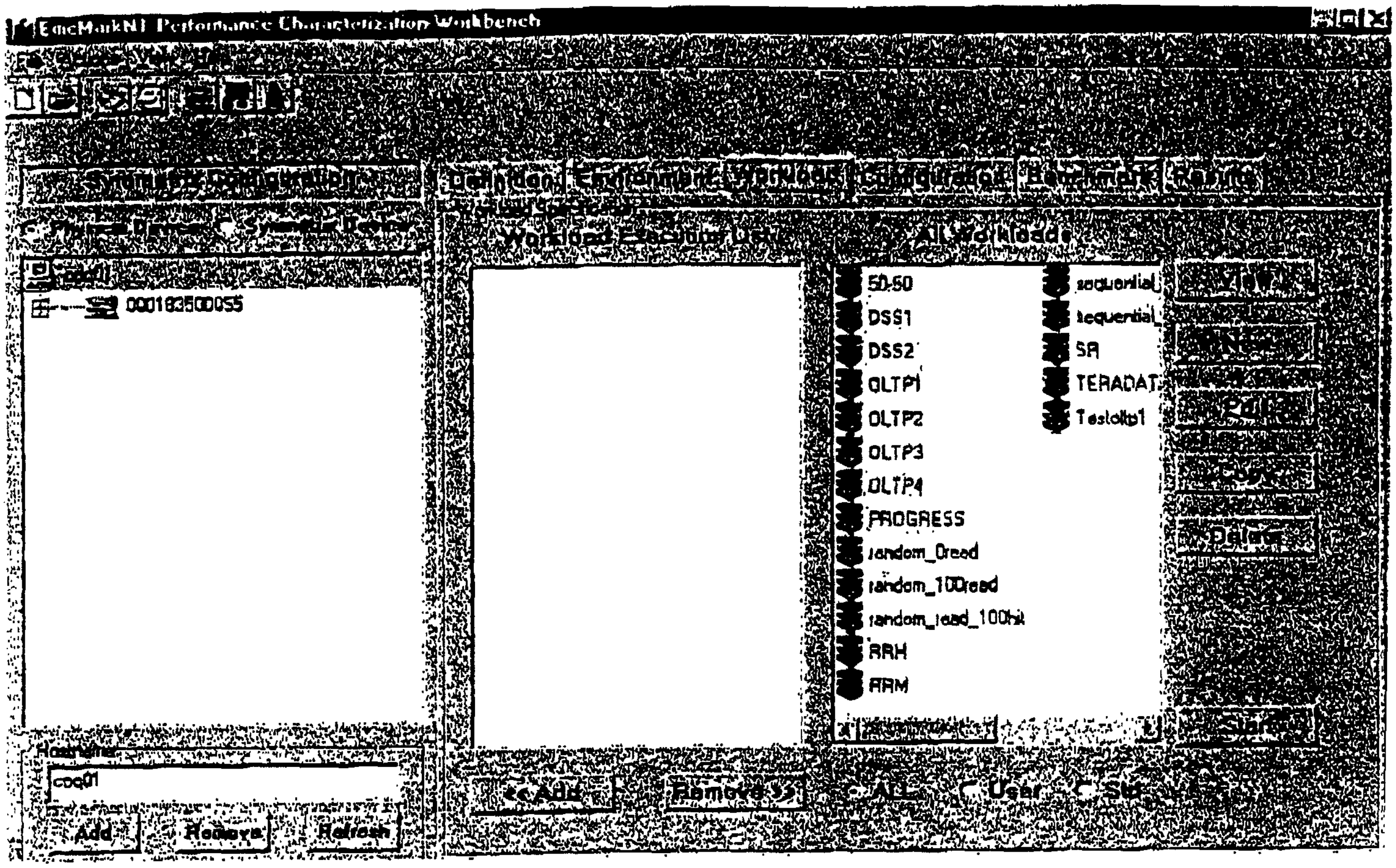


FIG. 9F

All - All available Workloads **User** – User Defined Workloads **Std** – Standard Workloads

View - Allows viewing of the detailed definition of aWorkloads
New - Brings up the Define Workloads form to define a new Workloads
Edit - Allows editing of User Defined Workloads
Copy - Copies the selected Workload into a new name, then brings up the Edit screen allowing edits to the new Workload.
Delete - Only for User Defined Workloads. Allows the deletion of a Workload.
Add - Moved the selected Workload over to the Workloads Execution List.
Remove- Removes item from the Workloads Execution List to the All Workloads List
Start - Brings up the Workloads Execution form to define and start the Workloads

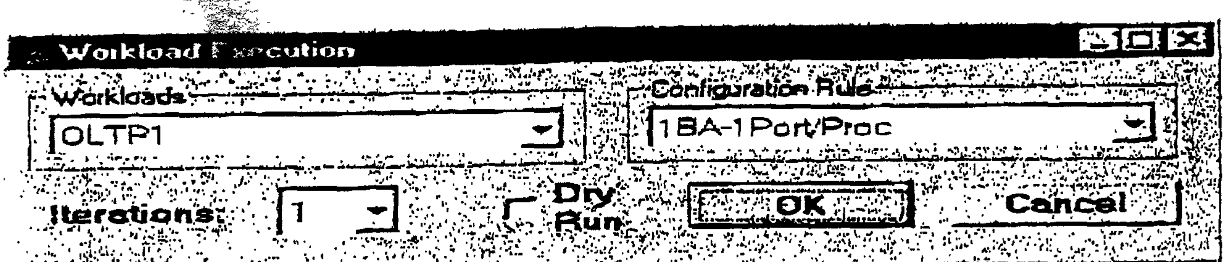


FIG. 9G

Iterations- The number of iterations the Workload should run for
Dry Run - Dry run will run through the scripts but not execute the Workload
OK - Will execute the Workload, bringing up the EmcMark Workload monitor window
Cancel - Will cancel the Workload execution

Response Time Workload

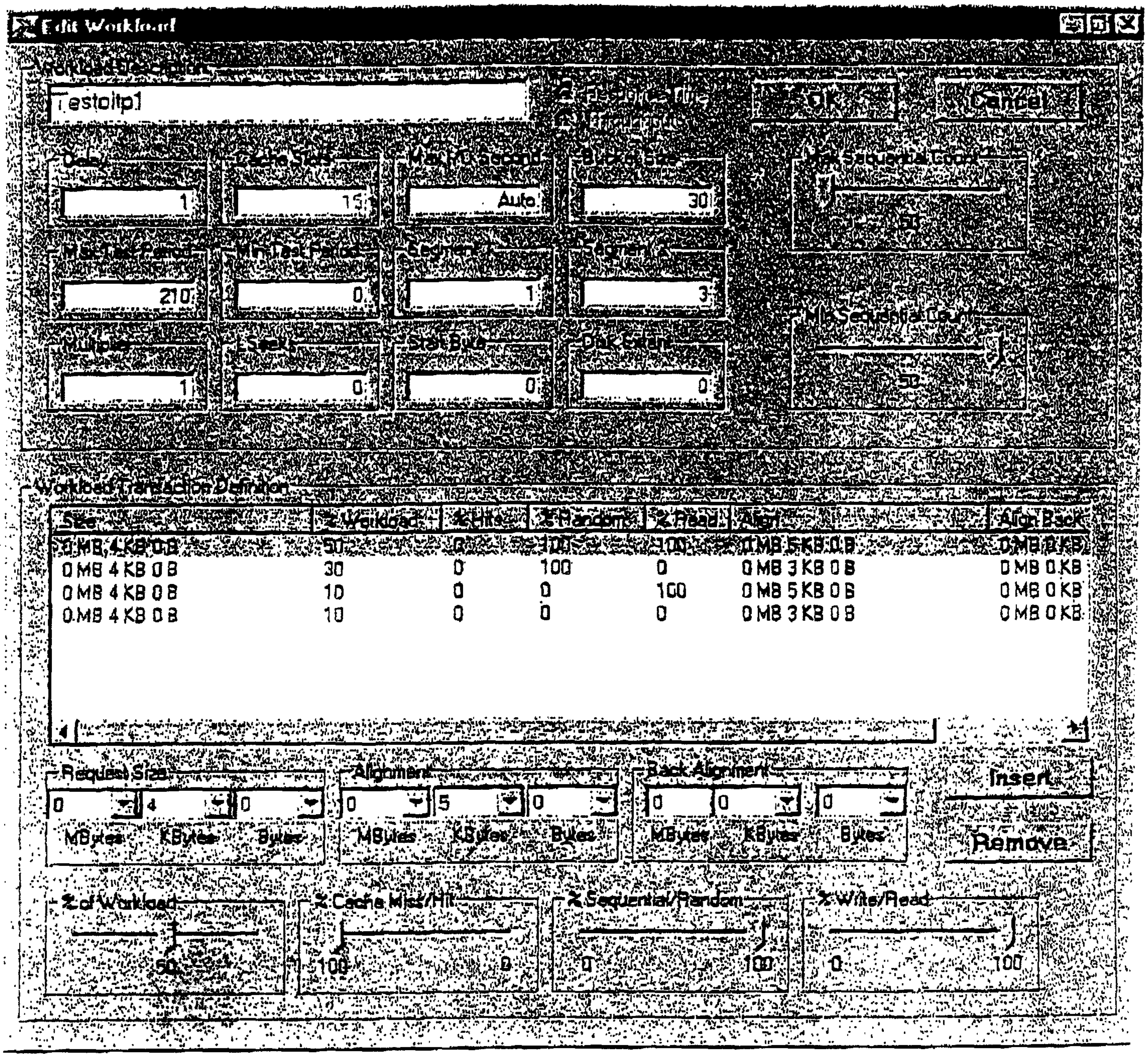


FIG. 9H

-Max Seq cannot be ϕ
-Min Seq - max is Max Seq selection

arrow Key - move by 10
-Page - move by 100

- Throughput Workload

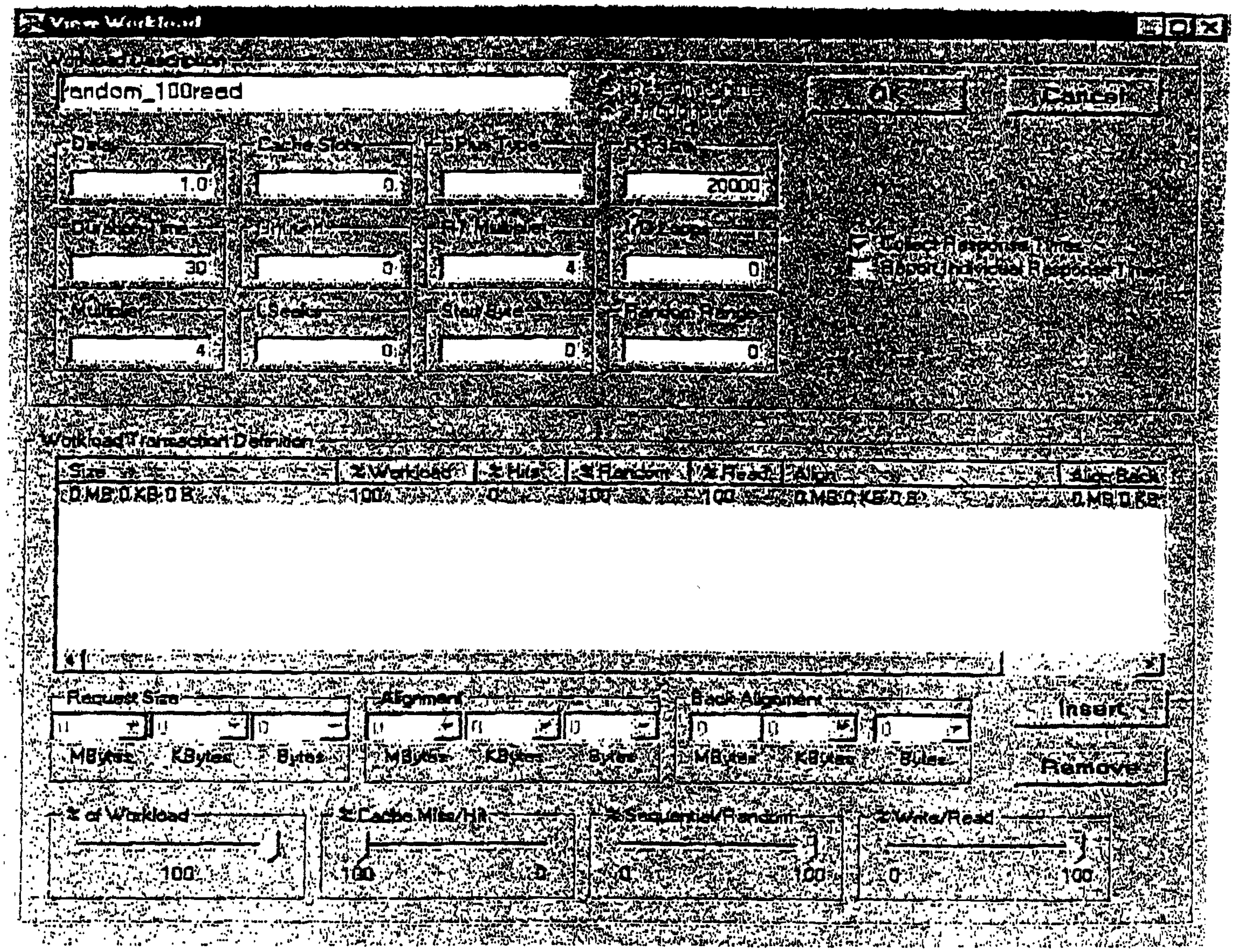


FIG 9I

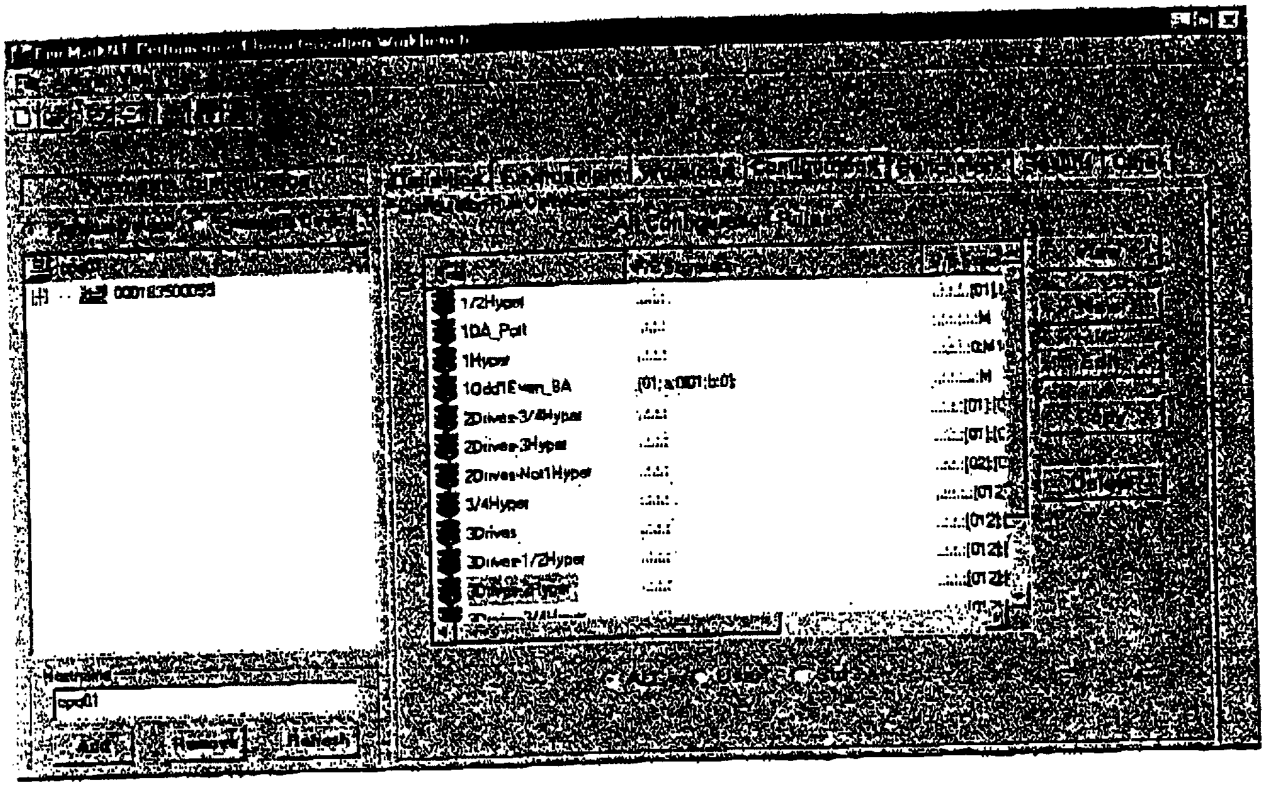


FIG. 9J

All - All available Rules **User** – User Defined Rules **Std** – Standard Rules

View - Allows viewing of the detailed definition of Rule
New - Brings up the Define Workloads form to define a new Rule
Edit - Allows editing of User Defined Rules
Copy - Copies the selected Rule into a new name, then brings up the Edit screen allowing edits to the new Rule.
Delete - Only for User Defined Rules. Allows the deletion of a Rule.

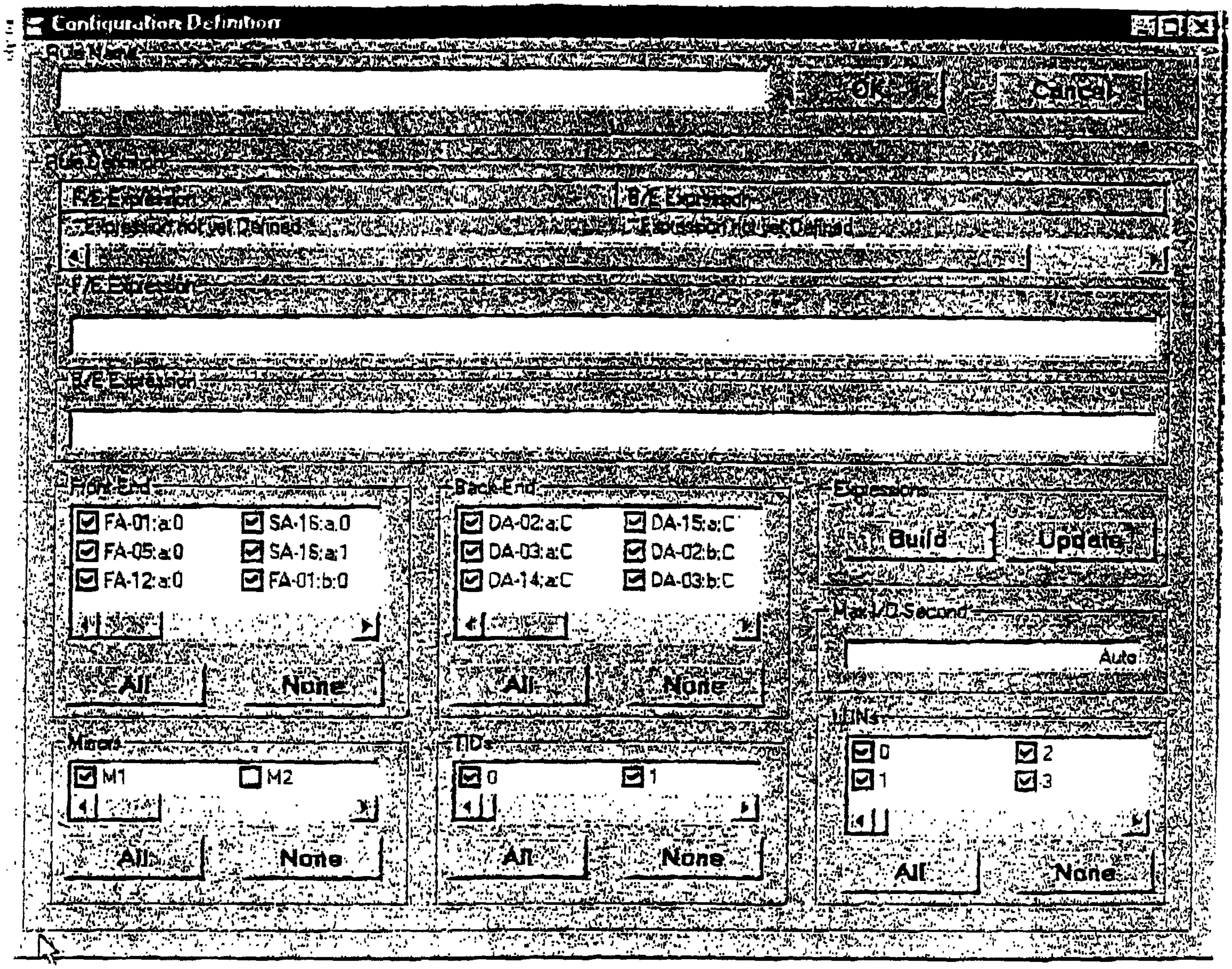


Fig 7K

Front End - BA/Processor/Port information pulled from SymAPI if Symmetrix connected
Back End - DA/Processor/Port information pulled from SymAPI if Symmetrix connected

Mirros. TIDs and Luns information pulled from SymAPI if Symmetrix connected

All/None buttons toggle checked boxes on or off.

Build Button – will built the expressions if the information has been downloaded from the Symmetrix. If no information is available then the expressions can be manually added to the F/E Expression and B/E Expression boxes.

Update Button – will update the F/E Expression and B/E Expression into the database.

OK will save the rule into the database
Cancel will terminate the definition

Benchmark Tab

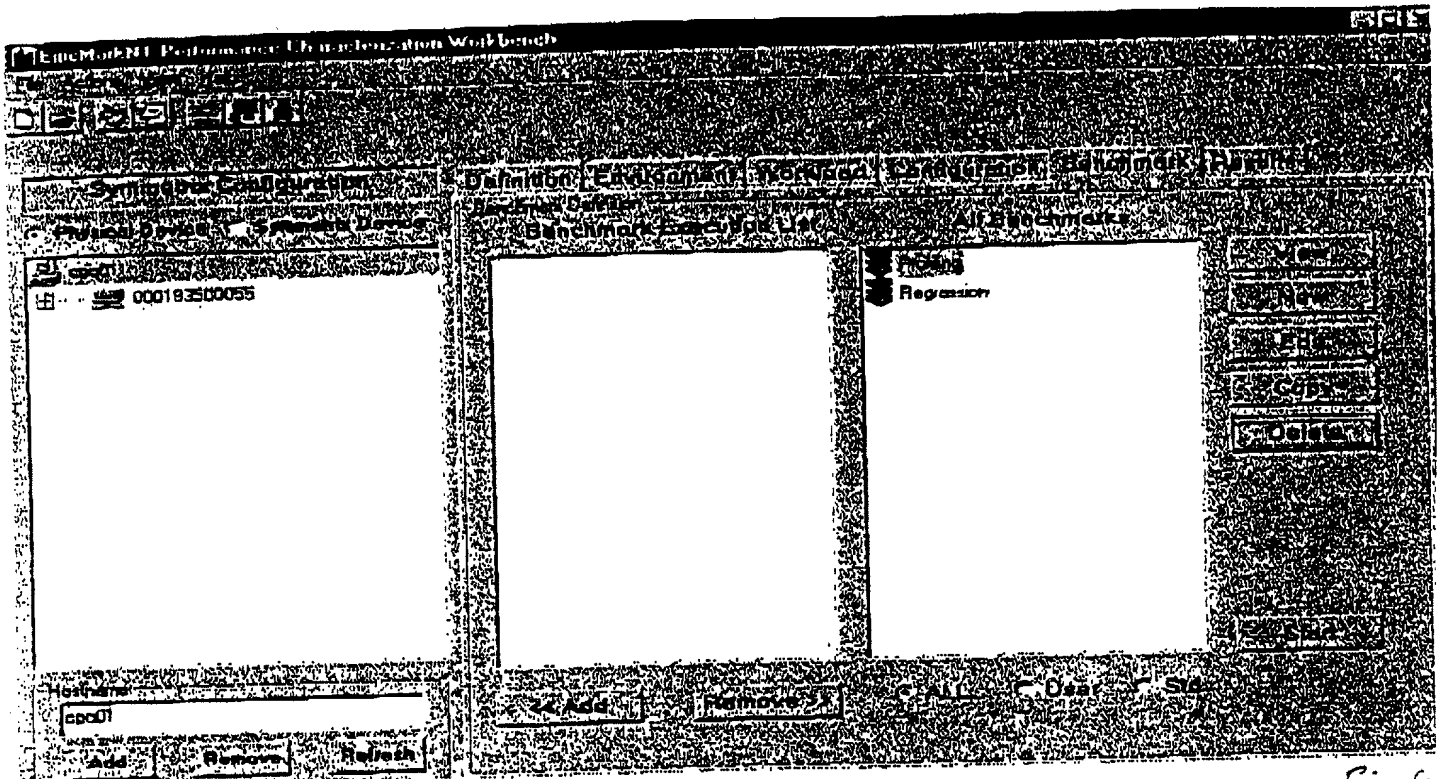


Fig 9

All - All available Benchmarks **User** – User Defined Benchmarks **Std** – Standard Benchmarks

View - Allows viewing of the detailed definition of a benchmark
New - Brings up the Define Benchmark form to define a new benchmark
Edit - Allows editing of User Defined Benchmarks
Copy - Copies the selected benchmark into a new name, then brings up the Edit screen allowing edits to the new benchmark.
Delete - Only for User Defined Benchmarks. Allows the deletion of a benchmark.
Add - Moved the selected benchmark over to the Benchmark Execution List. Only one Benchmark can be in the Execution list at a time
Remove- Removes item from the Benchmark Execution List to the All Benchmarks List
Start - Brings up the Benchmark Execution form to define and start the benchmark

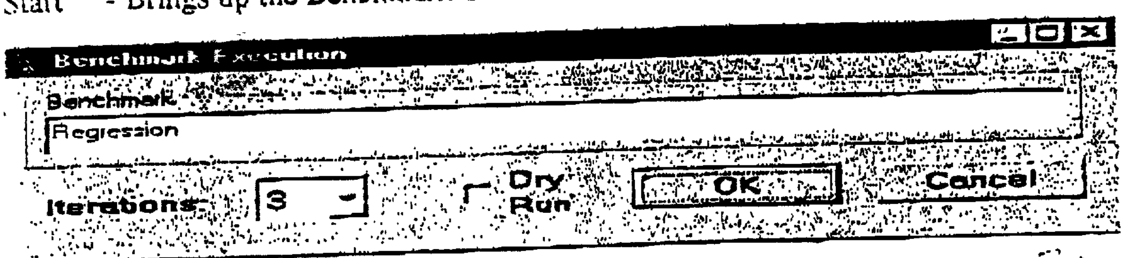


Fig 9M

Iterations- The number of iterations the benchmark should run for
Dry Run - Dry run will run through the scripts but not execute the benchmark
OK - Will execute the benchmark, bringing up the EmcMark Benchmark monitor window
Cancel - Will cancel the benchmark execution

Define Benchmark

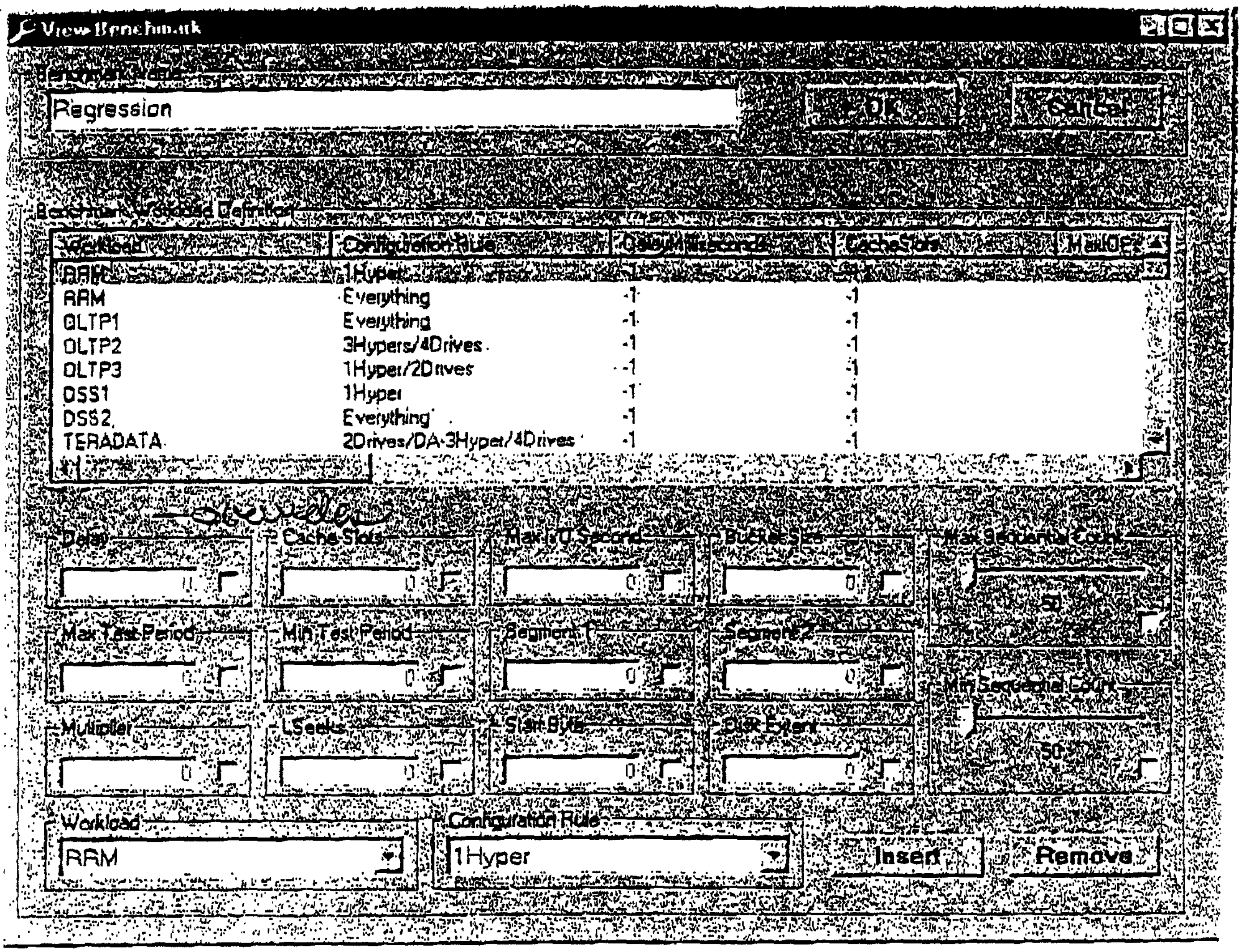


Fig. 9P

Results Tab

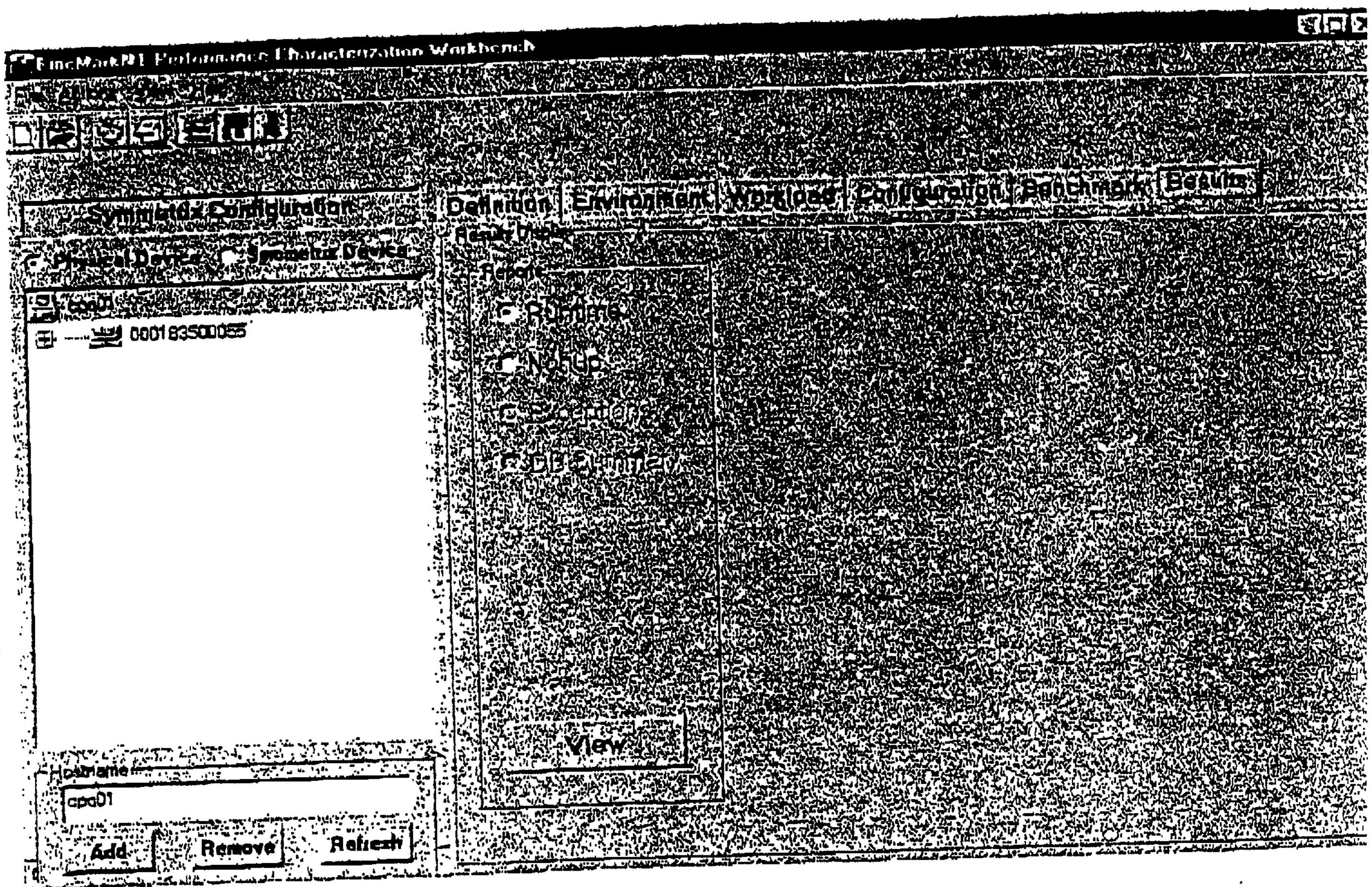


Fig 9C

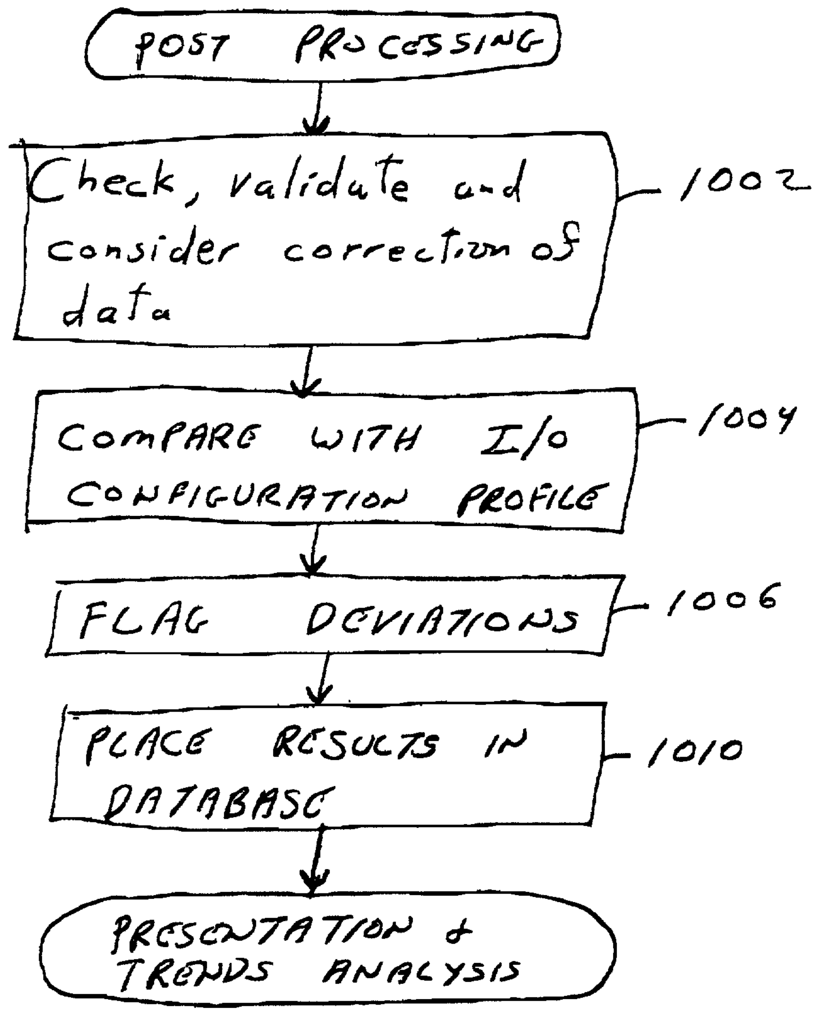

POST PROCESSING
Check, validate and consider correction of data
1002
COMPARE WITH I/O CONFIGURATION PROFILE
1004
FLAG DEVIATIONS
1006
PLACE RESULTS IN DATABASE
1010
PRESENTATION & TRENDS ANALYSIS

FIG. 10

Process Advanced Characterization Raw Data File

This function is used when your data format is not standard and you need to sort your data in the correct format for Splus to read the file.

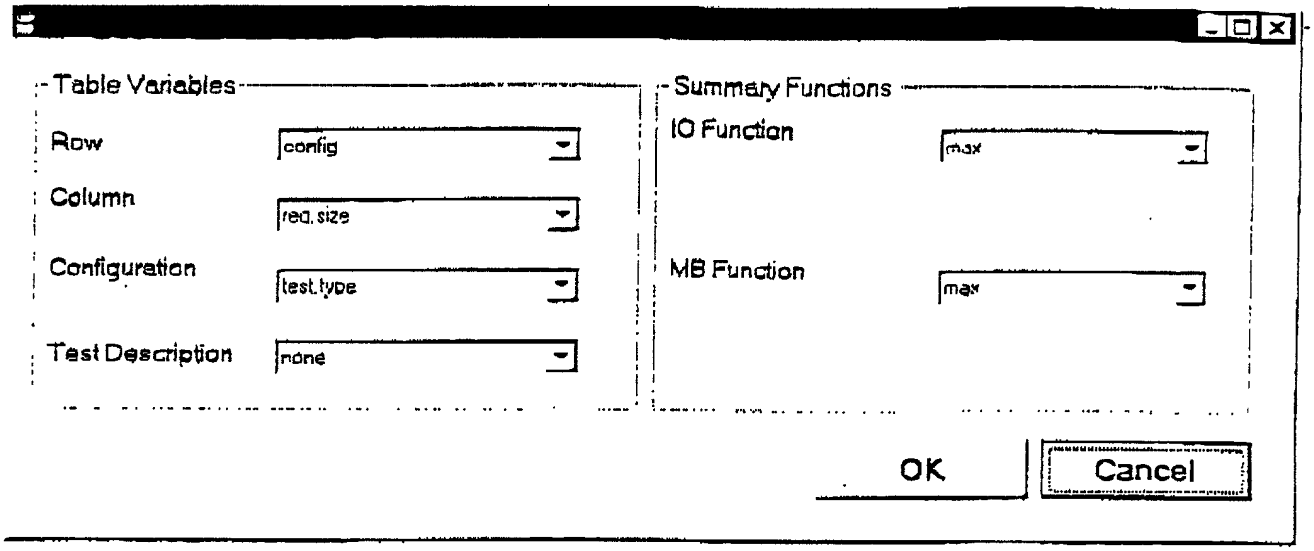


Fig. 10 A

1. Select CTRL A
2. The Advanced Characterization Window will appear
3. Select the correct row/column/configuration/test description options for your data
4. Select the summary functions for your data
5. Select OK
6. A Characterization file will be generated in the Post Processing Folder with the extension _adv.txt

Process Advanced DB Simulator Raw Data File

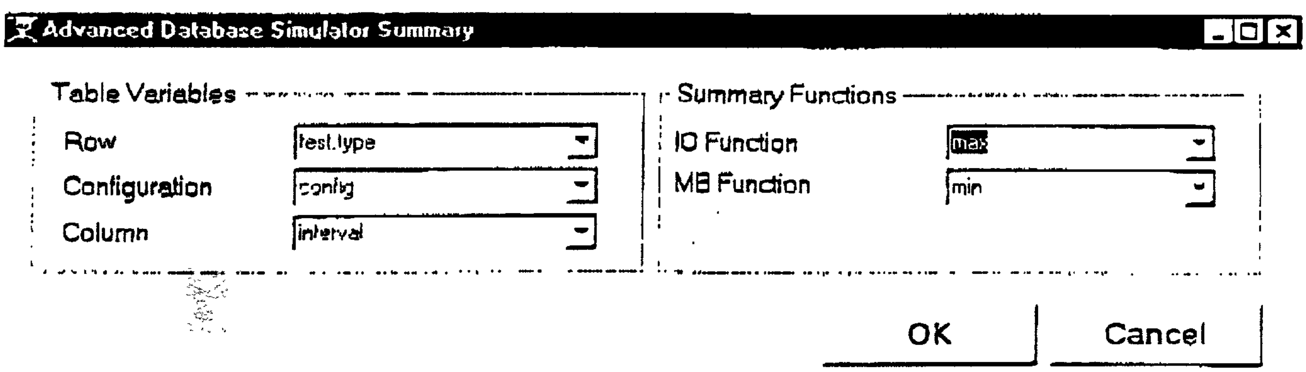


FIG. 10 B

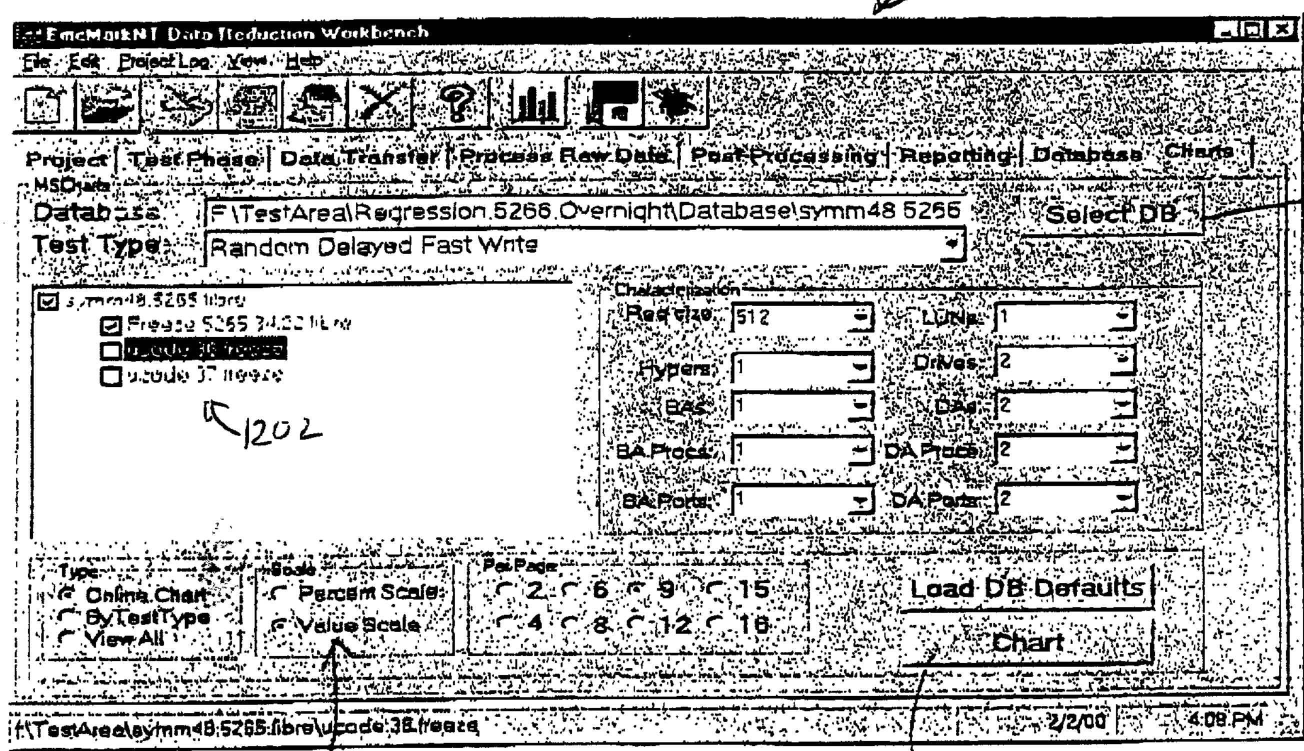
1. Select CTRL B
2. The Advanced DB Simulator Window will appear
3. Select the correct row/column/configuration/test description options for your data
4. Select the summary functions for your data
5. Select OK
6. A DB Simulator file will be generated in the Post Processing Folder with the extension _adv.txt

File Descriptions

File Name	Description	HighLights
Char.Summary	Summary file of each Characterization test broken down by iteration, test type, and configuration	
Char.Splus	Data file feed to Splus to create Characterization Objects	
Char.Errors	Characterization errors produced from processing the raw data files.	Message appears if error file exists.
SX.Summary	SX summary data broken down by iteration, test type and configruation.	
SX.Splus	Data file feed to Splus. Used with Char.Summary file to create Characterization Objects	
SX.Errors	SX errors from processing the raw data files	Message appears if error file exists.
DB.Table	Summary file of each DB Simulator test broken down by iteration, test type and configuration	
DB.Splus	Data file feed to Splus to create DBSimulator Objects	
DB.Errors	DB Simulator errors produced from processing the raw data files	Message appears if error file exists.
SX_DB.Summary	SX DB summary data broken down by iteration, test type and configuration.	
SX_DB.Splus	Data file feed to Splus. Used with DB.Splus file to create DBSimulator Objects	
SX_DB.Errors	SX_DB errors produced from processing the raw data files	Message appears if error file exists.
Cache Ratio Report	Report tracking the Cache ratio from the Sym and the processed data	Report name: "CacheRatioReport.txt" Located in the Raw Data folder Message appears if a report

Fig. 11

1208



120

1202

1206

1204

Fig. 12

Aug-17-00 01:45 From- T-828 P.28/29 F-755

COMBINED DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled GRAPHICAL USER INPUT INTERFACE FOR TESTING PERFORMANCE OF A MASS STORAGE SYSTEM, the specification of which:

[X] is attached hereto.
[] was filed on _ as Application Serial No. _ and was amended on ________________.
[] was described and claimed in PCT International Application No. ________________ filed on ________________ and as amended under PCT Article 19 on ________________.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose all information I know to be material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby appoint the following attorneys and/or agents to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

Gary A. Walpert, Reg. No. 26,098
Timothy A. French, Reg. No. 30,175
Penelope S. Wilson, Reg. No. 29,751
John Gunther, Reg. No. 26,175
Krishnendu Gupta, Reg. No. 37,977
Charles C. Winchester, Reg. No. 21,040
Frank P. Occhiuti, Reg. No. 35,306
William Clark, Reg. No. 29,523
Leanne J. Fitzgerald, Reg. No. 40,606

Address all telephone calls to GARY A. WALPERT at telephone number (617) 542-5070.

Address all correspondence to GARY A. WALPERT at:

FISH & RICHARDSON P.C.
225 Franklin Street
Boston, MA 02110-2804

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patents issued thereon.

Full Name of Inventor: MAUREEN A. LALLY

Inventor's Signature: Maureen A. Lally Date: 8/16/2000

Residence Address: 9 Maynard Road
Berlin, MA 01503

Citizenship: U.S.A.

Post Office Address: 9 Maynard Road
Berlin, MA 01503

Full Name of Inventor: KENNETH R. GOGUEN

Inventor's Signature: Kenneth R. Goguen Date: 8/16/2000

Residence Address: 9 Maynard Road
Berlin, MA 01503

Citizenship: U.S.A.

Post Office Address: 9 Maynard Road
Berlin, MA 01503

Full Name of Inventor: PETER C. LAUTERBACH

Inventor's Signature: Peter C. Lauterbach Date: 8/16/2000

Residence Address: 19 Town Line Road
Franklin, MA 02038

Citizenship: U.S.A.

Post Office Address: 19 Town Line Road
Franklin, MA 02038

20076486.doc

United States Patent & Trademark Office

Office of Initial Patent Examination -- Scanning Division

Application deficiencies were found during scanning:

☐ Page(s)__________ of ______________________________ were not present for scanning.
(Document title)

☐ Page(s)__________ of ______________________________ were not present for scanning.
(Document title)

☒ Scanned copy is best available. Drawings